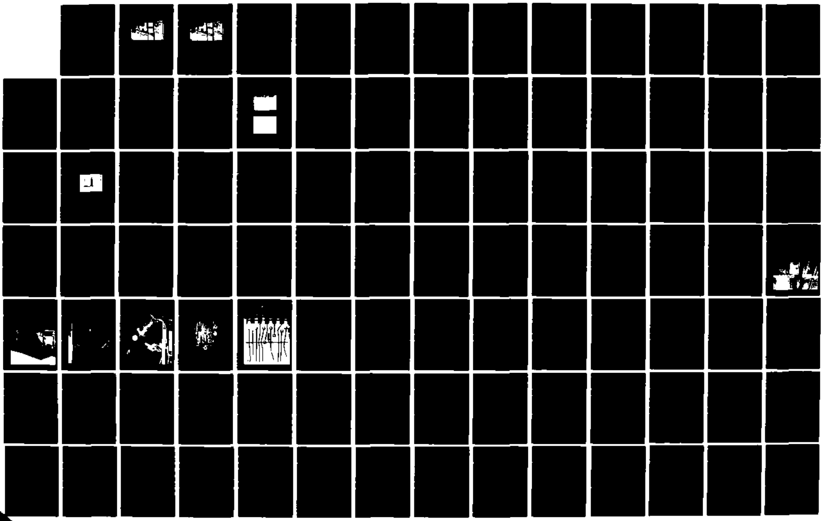
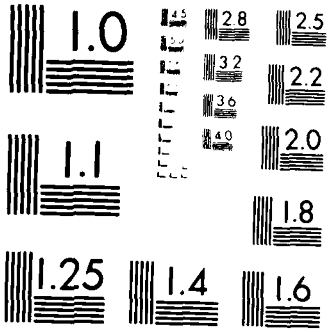
AD-A151 976 AUTOMATIC EVALUATION OF PRINTED WIRING BOARD SOLDER(U) 1/2
SCI SYSTEMS INC HUNTSVILLE AL 04 MAY 79
DAAK40-77-C-0105

UNCLASSIFIED F/G 9/5 NL





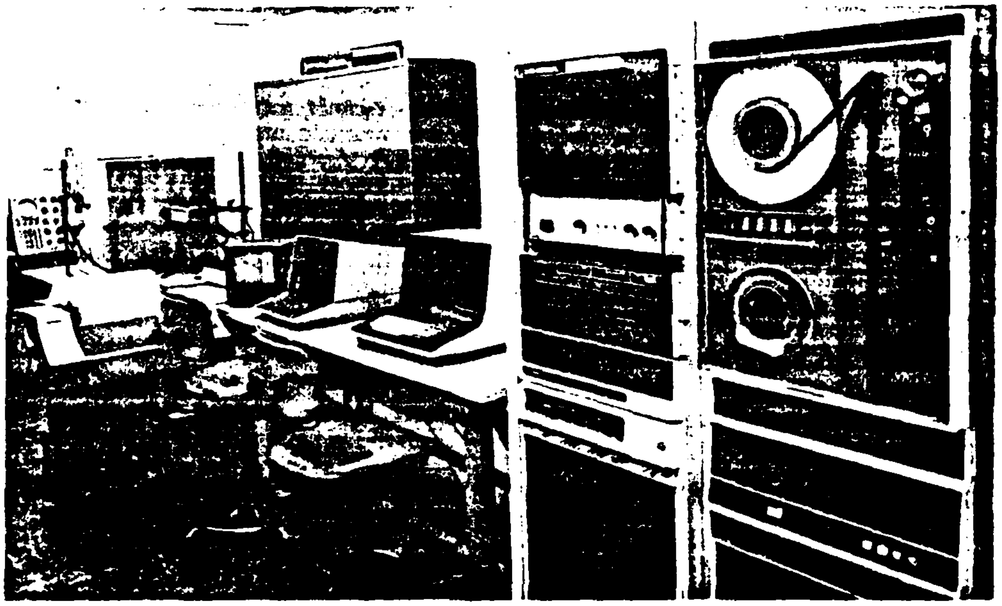
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

2

Proj 3169

final report

AD-A151 976



AUTOMATIC EVALUATION OF PRINTED WIRING BOARD SOLDER

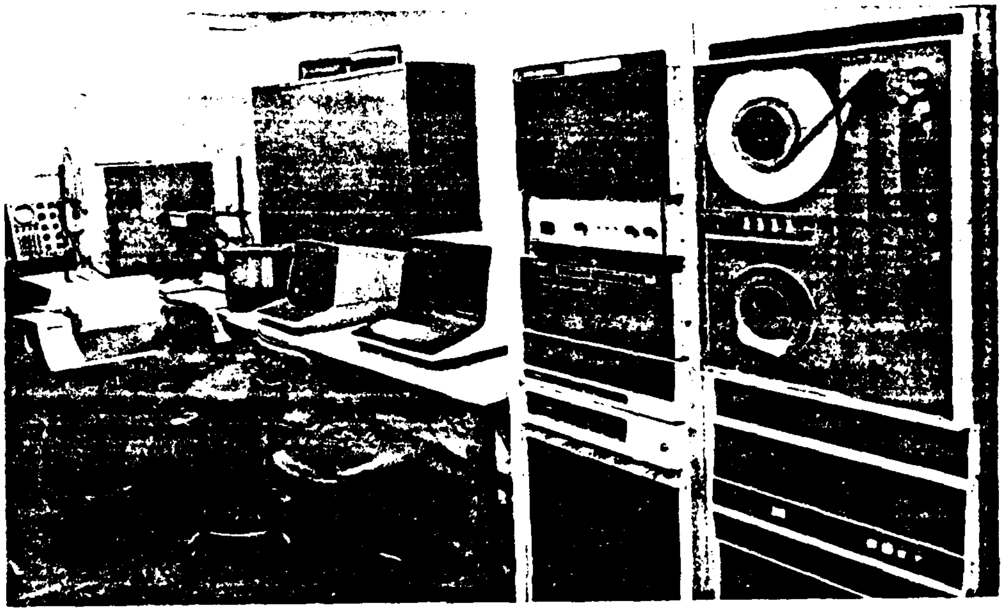
prepared for:

U.S. ARMY MISSILE RESEARCH AND
DEVELOPMENT COMMAND
REDSTONE ARSENAL, ALABAMA

CONTRACT NO. DAAK40-77-C-0105
4 MAY 1979

SCI SYSTEMS, INC.

final report



AUTOMATIC EVALUATION OF PRINTED WIRING BOARD SOLDER

prepared for:

U.S. ARMY MISSILE RESEARCH AND
DEVELOPMENT COMMAND
REDSTONE ARSENAL, ALABAMA

CONTRACT NO. DAAK40-77-C-0105
4 MAY 1979

ABSTRACT

This effort was expended to investigate the feasibility of using an optical scanning system to inspect printed circuit boards for defects. To implement this effort a Data General C/300 Eclipse was used interfacing with a Colorada Video digitizer and a Panasonic TV Camera/Monitor system. It was found that the system could operate well on the bare P.C. board without "pre-schooling" and would reveal imperfections in circuit lands, holes, and logos.

TABLE OF CONTENTS

I. SUMMARY

The expended effort was intended to determine how optical techniques might be applied to an automated fault detection board scanning system. The specific faults shown below were analyzed and a summary of the results are depicted in the Table below.

		Results
1.	Base electrical	
	a. Delamination	✓
	b. Measling	✓
2.	Component Alignment	X
3.	Conductor Flaw	✓
4.	Lifted Pad	✓
5.	Scratches	✓
6.	Surface Quality	✓
7.	Hole Registration	✓
8.	Legibility	✓

These analyses will be discussed in detail in the following section. In summation, all areas demonstrated a positive result with the one exception of the component alignment check. The approach selected eminated a flair for two-dimensional examination but had difficulty with any "3-D" requirements. In order to improve upon this, a higher resolution system (per unit time) would have to be used, such as a solid-state video psuedo stare-state mode monitor that would register a vertical line of images at a time, and choose these data points that are significant. This would involve a hardware pre-processing to prevent excessive CPU time. Direct benefits are obvious on a production scale with the currently developed software and a solid state scanner.

"Pre-schooling" would enhance the inspection process, however, the major thrust of the effort was directed toward a general workmanship specification approach. This was achieved with the one exception of component alignment checks. "Pre-schooling" has been shown to greatly improve this form of analysis.

II. INTRODUCTION

This document constitutes the final submittal for Contract DAAK40-77-C-0105.

Negotiations and final signing of the contract were completed on May 3, 1977. Since cost data is available at SCI using months of 4, 4, and 5 weeks per month during each quarter, the monthly progress and monthly financial reports were based on this same schedule.

Per the agreements prior to negotiation of the contract, SCI provided a computer system for use while this work was being conducted at SCI. A Data General C/300 Eclipse computer system was moved into the work area and checked for proper operation. Additional electrical service was provided to the lab in order to accommodate the computer system plus other normal laboratory equipment. The elements of the system are depicted in Appendix A.

An evaluation of available video digitizers resulted in a decision to purchase an image digitizer, Model 270A by Colorado Video, which was easier to interface and operate than others.

Other equipment in the system included a graphics monitor/terminal, a TV camera, and a video monitor. A paper-tape reader was also interfaced to the C300 computer. It was used to allow access of a wider variety of general support software than was available on magnetic tape.

The task of locating sources for board defects was conducted along with a review of board and solder specifications.

III. DISCUSSION

Implementation of Hardware

A standard unmapped real time disk operating system (RDOS Rev. 5) was installed in the computer and various service routines checked out. This was later replaced by INFOS. An illumination system was designed, which included a PC board mounting technique.

An illumination box was decigned to allow proper illumination of the printed circuit board under both diffuse and spot lighting conditions with a manually moveable mount for the board located on the illumination box. A printed wiring board was mounted on the outside of the box so that the camera, which is mounted on the opposite side, could see the board through holes in the box. Lamps were also mounted on the box such that either a spot can be placed on the board or scattered light from inside the box impinges on the board. The inside of the box is both diffuse and white to get maximum scattering of the light. A television camera was selected to serve as the primary sensor.

With the camera on, tests were .made on the preliminary software. These tests indicated that more resolution would be required than was available using the standard lens which came with the camera. An adaptor was made to allow the use of a photographic camera lens and extension tube. This increased the resolution significantly.

The camera and lens and two lamp sources were mounted to a wooden base, and the diffusion box was fastened to this base. With this system, the diffuse illumination was satisfactory. The specular illumination scheme did not produce the anticipated results. This was due to lack of reflectance of the board material and to a dullness of some of the solder on the sample board.

This condition is normally encountered in industry standards and while it is a negative result, it should be noted as an approach not to be attempted.

For convenience and clarity, the technical requirements that guided this effort are included in the following section.

TECHNICAL REQUIREMENT NO. 6148

1. SCOPE

1.1 GENERAL

The purpose of this project was to determine methods and technology of inspecting printed wiring boards at the bare board or populated/soldered level by an automatic video scanning method and to demonstrate the feasibility of such a process as a cost reducing method for the determination of an optimum methodology for the location/evaluation of specific classes of [illegible] defects.

2. APPLICABLE DOCUMENTS

2.1

MIL-STD-454 - Standard General Requirements for Electronic Equipment.

2.2

MIL-STD-275 - Printed Wiring for Electronic Equipment.

2.3

MIL-STD-1495 - Multilayer Printed Wiring Boards for Electronic Equipment.

2.4

MIL-S-45743 - Soldering, Manual Type, High Reliability.

2.5

MIL-S-46844 - Solder Bath Soldering of Printed Wiring Assemblies, Automatic Machine Type.

3. REQUIREMENTS

3.1 TECHNICAL EFFORT

The contractor provided the personnel, materials, equipment and expertise to perform the tasks as hereinafter described.

3.2 PLANNING

Technical effort conducted under these technical requirements were planned in terms of technical milestones to be accomplished in pursuit of the program. Each milestone indicated completion of a significant portion of the effort. A milestone chart showing the entire program was prepared by the contractor and submitted in an oral presentation within thirty (30) days after the contract award. The government reserved seven (7, days after receipt to review and concur/nonconcur in the detailed proposed plan; if the contractor did not receive notification within seven (7) days, he was to assume concurrence. The milestone chart and the government concurrences/nonconcurrences were documented.

3.3 DETAILS OF EFFORT TO BE PERFORMED

3.3.1

The contractor demonstrated an image processing system consisting of a contractor supplied computer and the following items obtained and assembled as a part of this contract:

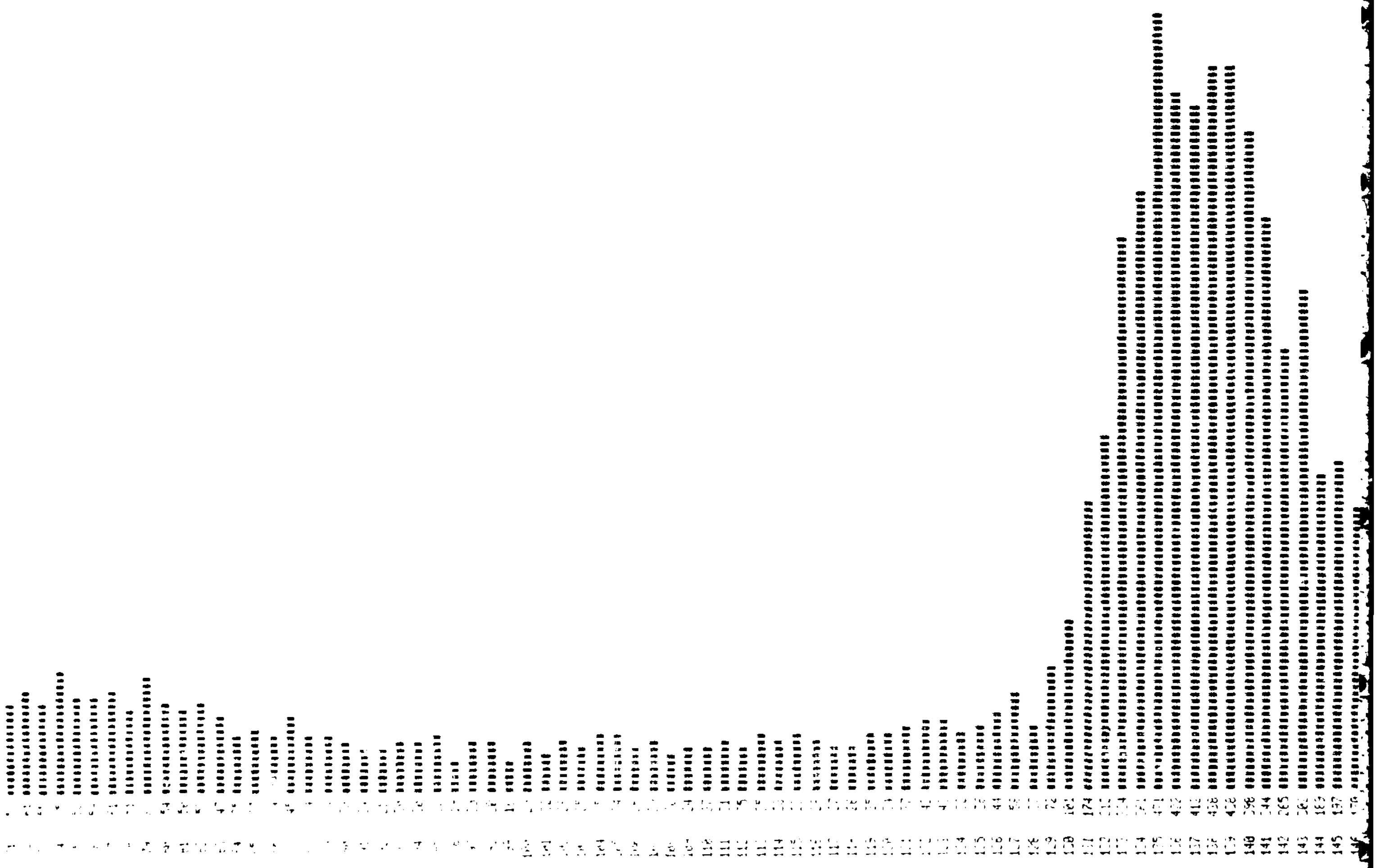
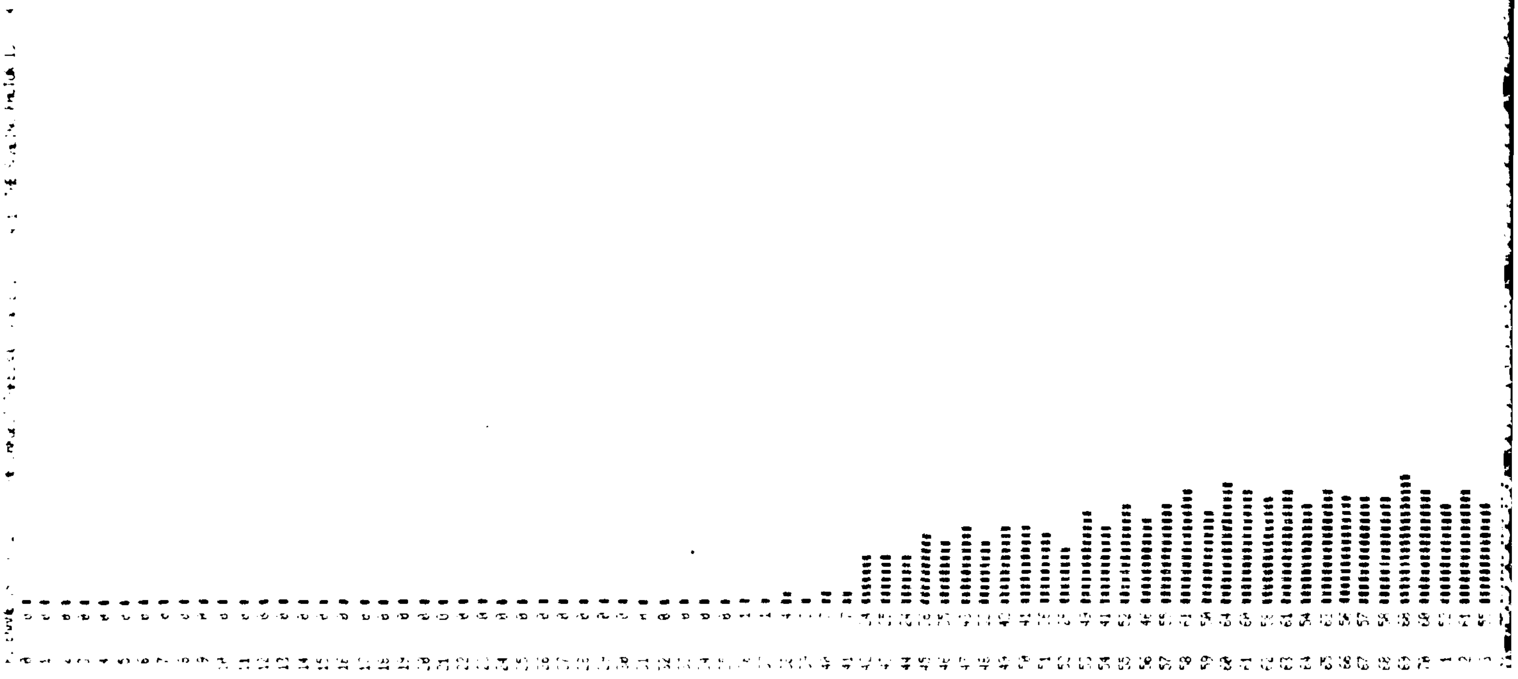
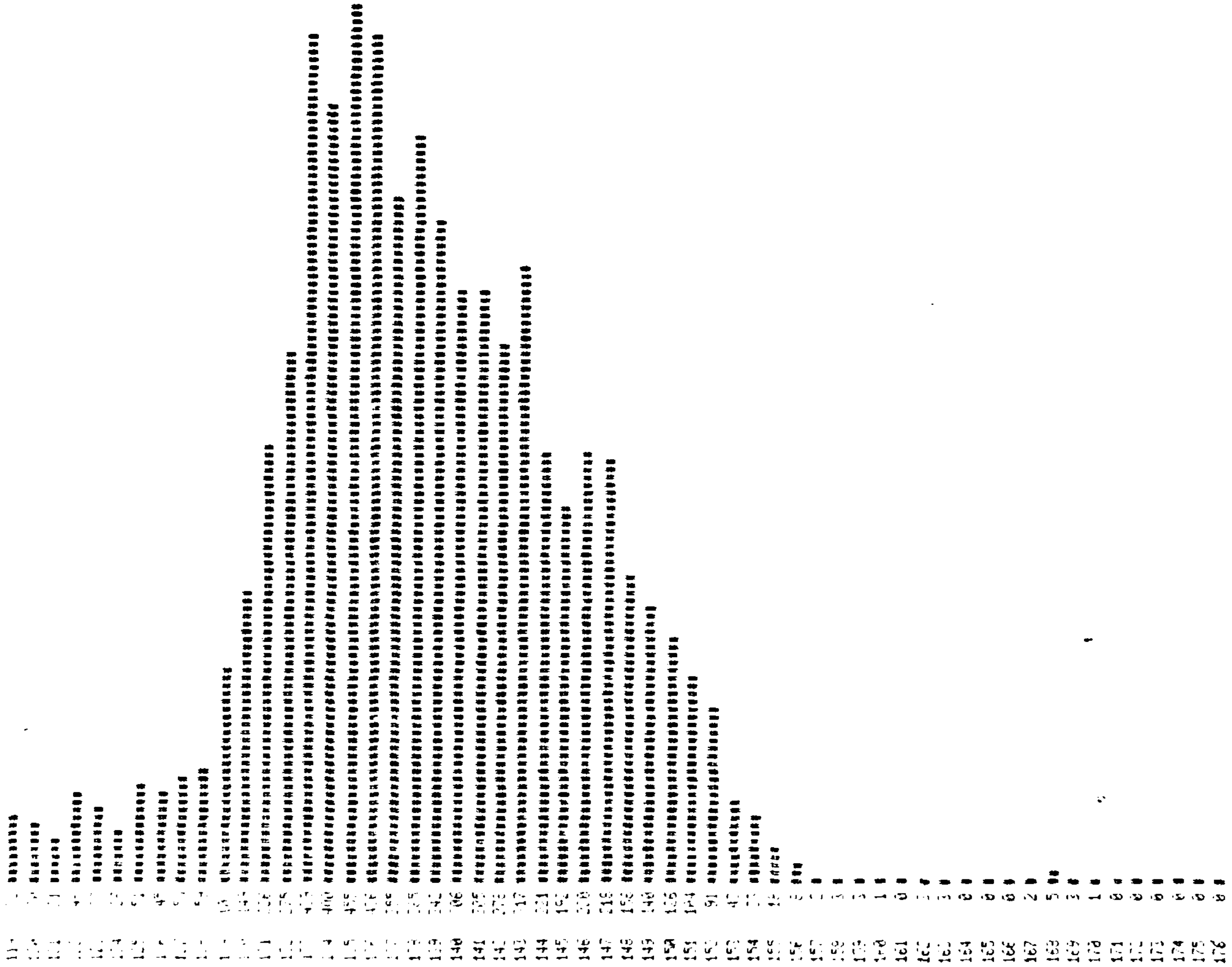


FIGURE 6 PPTHIST OUTPUT FOR BOARD IN FIGURE 2

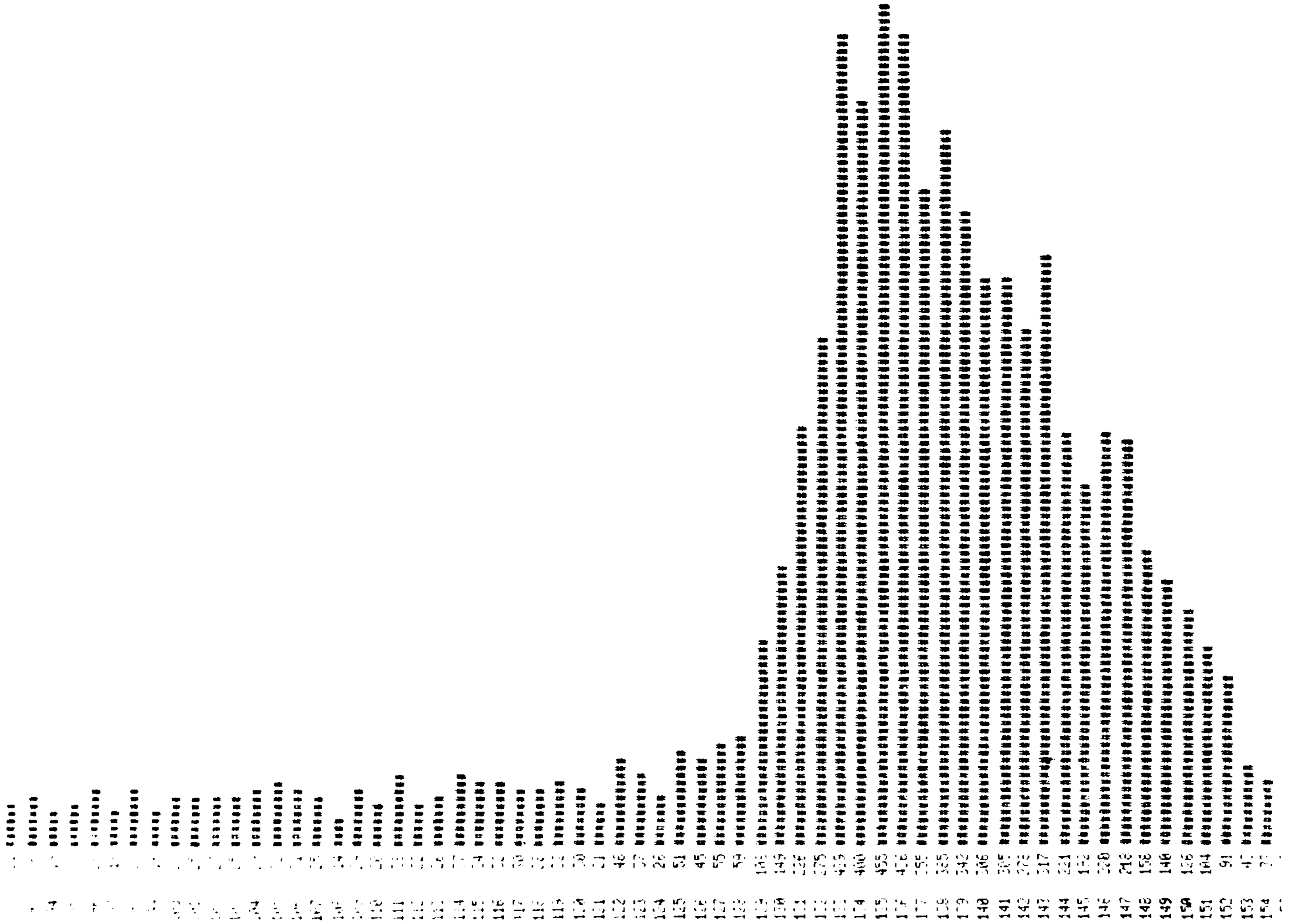


3



4 of 4.

3



FOR BOARD IN FIGURE 1

3 of 4

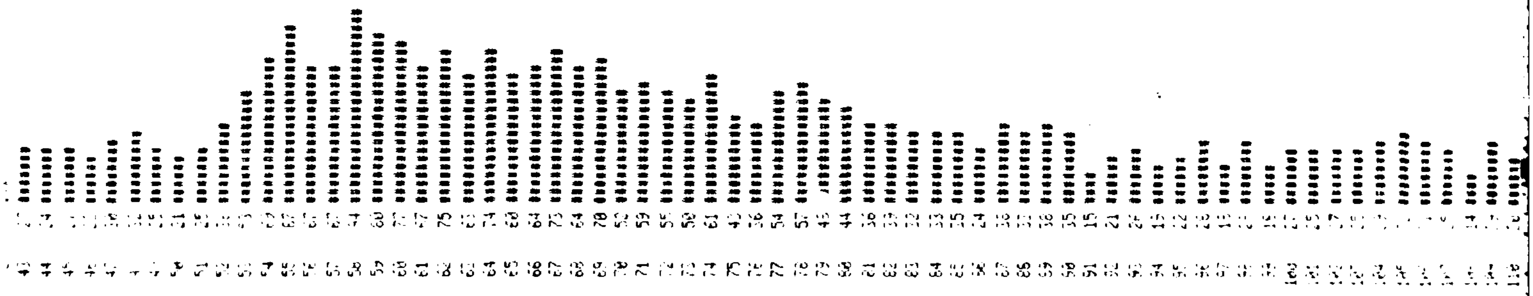


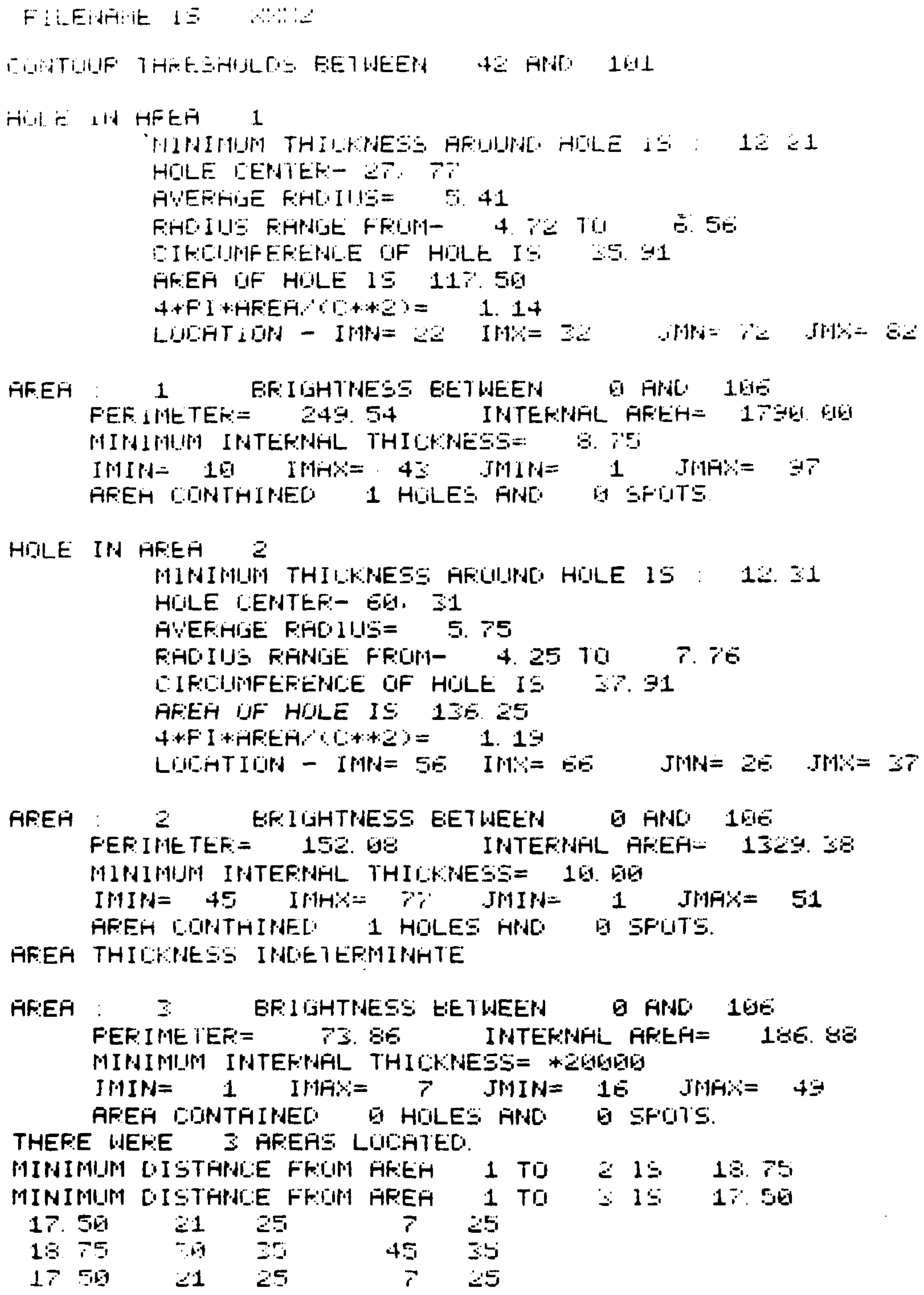
FIGURE 5 PRTHIST OUTPUT FOR BOARD IN FIGURE 1

2 of 4

16

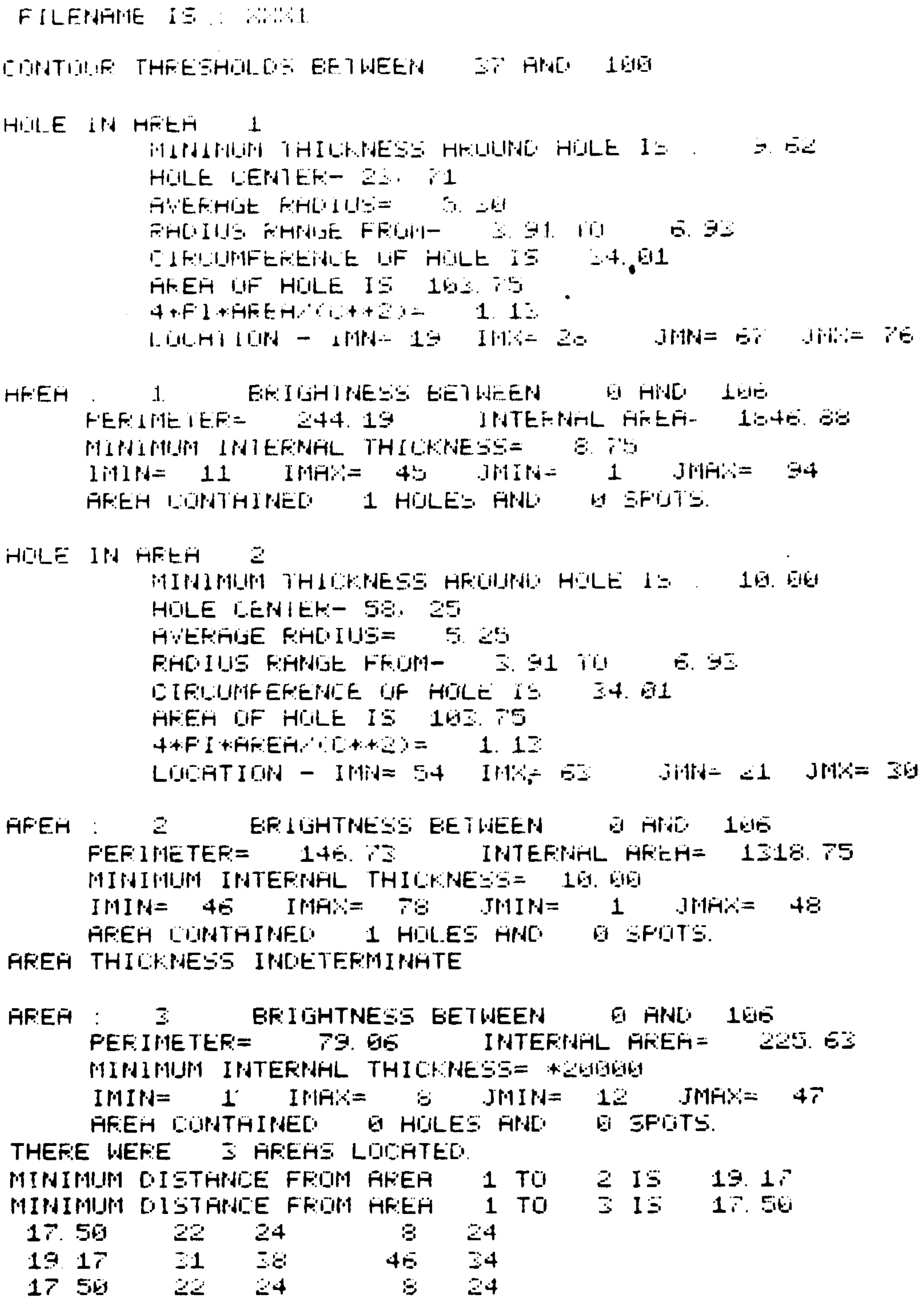
1 of 4

```
FILENAME IS  XXX1      THE LARGEST THRESHOLD VALUE IS     455  THE SCALING FACTOR IS    4
0       0 #
1       0 #
2       0 #
3       0 #
4       0 #
5       0 #
6       0 #
7       0 #
8       0 #
9       0 #
10      0 #
11      0 #
12      0 #
13      0 #
14      0 #
15      0 #
16      0 #
17      0 #
18      0 #
19      0 #
20      0 #
21      0 #
22      0 #
23      0 #
24      0 #
25      0 #
26      0 #
27      0 #
28      0 #
29      0 #
30      0 #
31      0 #
32      0 #
33      1 #
34      1 #
35      7 ##
36      5 ##
37     15 ####
38     10 ###
39     15 ####
40     21 ######
41     27 #######
42     19 #####
43     27 #######
44     24 #######
45     27 #######
46     22 ######
47     30 ########
48     32 #########
49     25 #######
50     21 ######
51     25 #######
52     36 ##########
53     53 ##############
54     69 ##################
55     87 ######################
56     67 #################
57     67 #################
```



```
 FILENAME IS   [illegible]

CONTOUR THRESHOLDS BETWEEN    42 AND  101

HOLE IN AREA    1
          MINIMUM THICKNESS AROUND HOLE IS :   12.21
          HOLE CENTER- 27, 77
          AVERAGE RADIUS=    5.41
          RADIUS RANGE FROM-    4.72 TO     6.56
          CIRCUMFERENCE OF HOLE IS    35.91
          AREA OF HOLE IS  117.50
          4*PI*AREA/(C**2)=    1.14
          LOCATION - IMN= 22  IMX= 32     JMN= 72  JMX= 82

AREA :     1       BRIGHTNESS BETWEEN      0 AND  106
     PERIMETER=    249.54       INTERNAL AREA=  1790.00
     MINIMUM INTERNAL THICKNESS=    8.75
     IMIN=  10    IMAX=  43    JMIN=    1    JMAX=  97
     AREA CONTAINED    1 HOLES AND    0 SPOTS.

HOLE IN AREA    2
          MINIMUM THICKNESS AROUND HOLE IS :   12.31
          HOLE CENTER- 60, 31
          AVERAGE RADIUS=    5.75
          RADIUS RANGE FROM-    4.25 TO     7.76
          CIRCUMFERENCE OF HOLE IS    37.91
          AREA OF HOLE IS  136.25
          4*PI*AREA/(C**2)=    1.19
          LOCATION - IMN= 56  IMX= 66     JMN= 26  JMX= 37

AREA :     2       BRIGHTNESS BETWEEN      0 AND  106
     PERIMETER=    152.08       INTERNAL AREA=  1329.38
     MINIMUM INTERNAL THICKNESS=  10.00
     IMIN=  45    IMAX=  77    JMIN=    1    JMAX=  51
     AREA CONTAINED    1 HOLES AND    0 SPOTS.
AREA THICKNESS INDETERMINATE

AREA :     3       BRIGHTNESS BETWEEN      0 AND  106
     PERIMETER=     73.86       INTERNAL AREA=    186.88
     MINIMUM INTERNAL THICKNESS= *20000
     IMIN=    1    IMAX=    7    JMIN=  16    JMAX=  49
     AREA CONTAINED    0 HOLES AND    0 SPOTS.
THERE WERE    3 AREAS LOCATED.
MINIMUM DISTANCE FROM AREA    1 TO    2 IS    18.75
MINIMUM DISTANCE FROM AREA    1 TO    3 IS    17.50
 17.50     21    25       7   25
 18.75     30    35      45   35
 17 50     21    25       7   25
```

FIGURE 4 PRINT OF CHARACTERISTICS OF BOARD IN FIGURE 2



```
 FILENAME IS : [illegible]

CONTOUR THRESHOLDS BETWEEN    37 AND  100

HOLE IN AREA    1
             MINIMUM THICKNESS AROUND HOLE IS :     9.62
             HOLE CENTER- 23, 71
             AVERAGE RADIUS=    5.50
             RADIUS RANGE FROM-    3.91 TO     6.93
             CIRCUMFERENCE OF HOLE IS    34.01
             AREA OF HOLE IS  103.75
             4*PI*AREA/(C**2)=    1.13
             LOCATION - IMN= 19  IMX= 26     JMN= 67  JMX= 76

AREA :    1       BRIGHTNESS BETWEEN     0 AND  106
       PERIMETER=    244.19       INTERNAL AREA-  1646.88
       MINIMUM INTERNAL THICKNESS=    8.75
       IMIN=  11    IMAX=  45    JMIN=   1    JMAX=  94
       AREA CONTAINED    1 HOLES AND    0 SPOTS.

HOLE IN AREA    2
             MINIMUM THICKNESS AROUND HOLE IS :   10.00
             HOLE CENTER- 58, 25
             AVERAGE RADIUS=    5.25
             RADIUS RANGE FROM-    3.91 TO     6.93
             CIRCUMFERENCE OF HOLE IS    34.01
             AREA OF HOLE IS  103.75
             4*PI*AREA/(C**2)=    1.13
             LOCATION - IMN= 54  IMX= 63     JMN= 21  JMX= 30

AREA :    2       BRIGHTNESS BETWEEN     0 AND  106
       PERIMETER=    146.73       INTERNAL AREA=  1318.75
       MINIMUM INTERNAL THICKNESS=  10.00
       IMIN=  46    IMAX=  78    JMIN=   1    JMAX=  48
       AREA CONTAINED    1 HOLES AND    0 SPOTS.
AREA THICKNESS INDETERMINATE

AREA :    3       BRIGHTNESS BETWEEN     0 AND  106
       PERIMETER=     79.06       INTERNAL AREA=   225.63
       MINIMUM INTERNAL THICKNESS= *20000
       IMIN=   1    IMAX=    8    JMIN=  12    JMAX=  47
       AREA CONTAINED    0 HOLES AND    0 SPOTS.
THERE WERE    3 AREAS LOCATED.
MINIMUM DISTANCE FROM AREA    1 TO    2 IS    19.17
MINIMUM DISTANCE FROM AREA    1 TO    3 IS    17.50
  17.50      22    24          8    24
  19.17      31    38         46    34
  17 50      22    24          8    24
```

FIGURE 3   PRINT OF CHARACTERISTICS OF BOARD IN FIGURE 1

A specific sequence of operation will appear as follows:

1. The DOTMAT Routine is called and responds with the question "File Name?" which directs the entry of the desired data file. This data file is acquired by use of live scan using routine SCAN 2. The name of the file is arbitrary but must not exceed 10 characters in length.

2. The DOTMAT Routine provides the information to define the total number of detected points. This information can be used to compare with the expected norm, or to establish a norm. Also provided by DOTMAT is visual image of the area being scanned. PRTMAT is a companion routine that outputs on the line printer when a hard copy is required.

3. HISTOGRAM is run on the data file created by SCAN 2 and displays the information needed to select upper and lower thresholds to be used when/if PCCARD is used. If Step 1 is used on a test card and the printout "Probable Error on Board" appears, then Steps 2 and 3 apply. Otherwise, the test series would complete with Step 1. The CRT output of HISTOGRAM is shown in Figure 7.

4. PCCARD is activated by a series of inputs:
   a. "Print or Type" Enter P or T
   b. "Live Scan?" Enter Y or N
   c. "Enter Filename"
   d. File is opened and question displayed "Smooth Data?" Y or N
   e. Thresholds are defined and are optional; override entry is provided.

The output information provided includes the minimum distance from exterior and interior surfaces of the solder lands, and the data displayed by Figure 3.

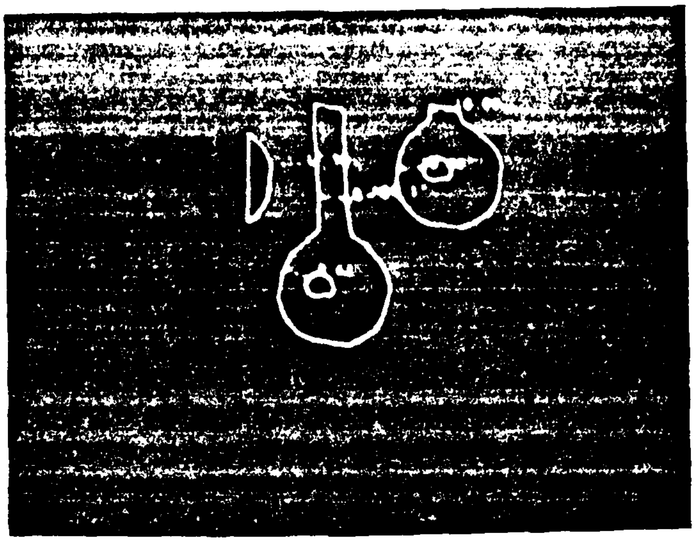


FIGURE 1 CRT DISPLAY OF OFF-CENTER HOLE

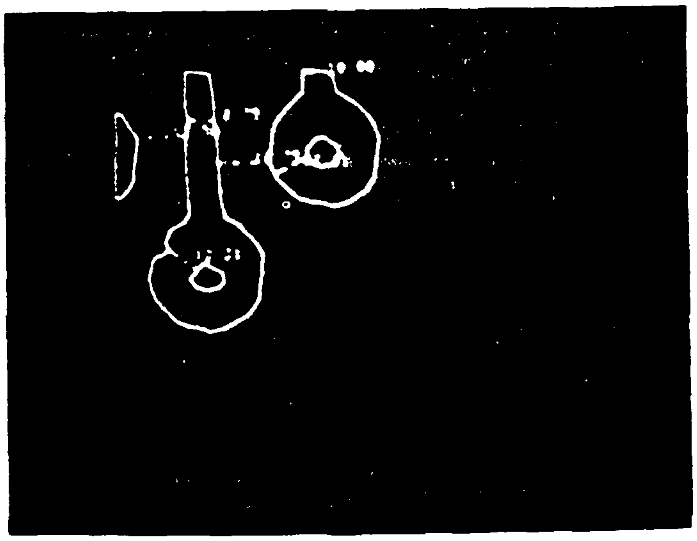


FIGURE 2 CRT DISPLAY OF DEFECT

System Operation

Operating the system under INFOS the operator first calls for the PCBRD directory. Then he has a choice of several analytical routines to use depending on the nature of the inspection. First a "standard" is recorded of a normal board and stored. The test boards are scanned and compared. The recommended procedure is to use the HISTOGRAM Routine to determine the upper and lower threshold limits, and the DOTMAT Routine to establish the number of points detected. This can then be followed by PCCARD to analyze the nature of the card in question in regard to a given spot. This above procedure is presented in more detail next.

A printed circuit board should be tested in the following manner. A sample card is pulled from a production run. The board is scanned with TV monitor system and the digitized data is compared with the standard. These two scans can be given arbitrary file names. In this example, the scans are given the names XXX2 for the standard, and XXX1 for the sample board. If more than a 1% difference in the number of detected points occurs, an error message will appear as "probable Error on Board". If not "Board OK" will appear. Following this if further checks are needed, then the HISTOGRAM or PRTHIST routines are employed to determine the threshold settings to use. Next, the PCCARD Routine is implemented and will call in the data file and will determine the minimum distance across any PC land and will display it on the Tektronix memory terminal. (See Figures 1 & 2). Also, Figures 3 & 4 depict information that can be observed in hard copy or CRT display. In this copy, the error shown is an off-center hole in a PC land. This is visible in Figure 1. The units displayed are mils. PRTHIST depicts a plot of the relative rate of occurrance of light levels on an arbitrary scale of 0 to 256. The routine HISTOGRAM displays the same plot on the CRT terminal, but does not display the threshold card scale factor data. The output of PRTHIS is displayed in Figures 5 & 6 for the boards in Figures 1 & 2.

4.3

This final report in narrative form was delivered. This report summarizes the work accomplished under this contract. This report fully documents the equipment, its interfacing, and programming manuals required to duplicate and utilize the demonstrator scanner for inspecting printed wiring boards within a production environment.

3.3.3

Printed circuit boards with samples of several levels of the defects under consideration were obtained. A rating system was devised to establish the severity of each defect and to allow a decision as to acceptable/non-acceptable for each type defect.

3.3.4

Methods of detecting and evaluating printed-circuit board defects were devised and tested using the system of 3.3.1. Implementation was in software wherever possible.

3.3.5

Based on the above, this report was prepared containing:

(A) Discussion of the defects and related scoring system developed in 3.3.3.

(B) Discussion and explanation of the detection methods developed in 3.3.4.

4. DOCUMENTATION

4.1

Data delivered, schedule of delivery, and distribution requirements are specified on DD Form 1423.

4.2

In addition, the contractor reviewed the progress of the work to the government at mutually agreed times and places. The number of such reviews held at places other than the contractor's facilities did not exceed three (3).

(A) Input Imaging Device with computer interface containing vidicon TV camera, scan control, analog/digital converter, computer bus interface electronics.

(B) Graphics display terminal capable of accepting images from the computer, storing the image, and displaying the image with several shades of gray.

(C) Mechanical system capable of holding and manipulating printed circuit boards under manual control for placement under the imaging and illumination system.

(D) Image processing software capable of controlling and operating the Input Imaging Device in order to bring images into the computer memory along with special processing of these images to include thresholding and variation of number of resolution elements.

3.3.2

Board Parameters evaluated include:

(A) Rough or ragged edges

(B) Rough, burred holes

(C) Haloing extending between conductors

(D) Measling

(E) Delamination

(F) Localized blistering

(G) Exposed glass weave

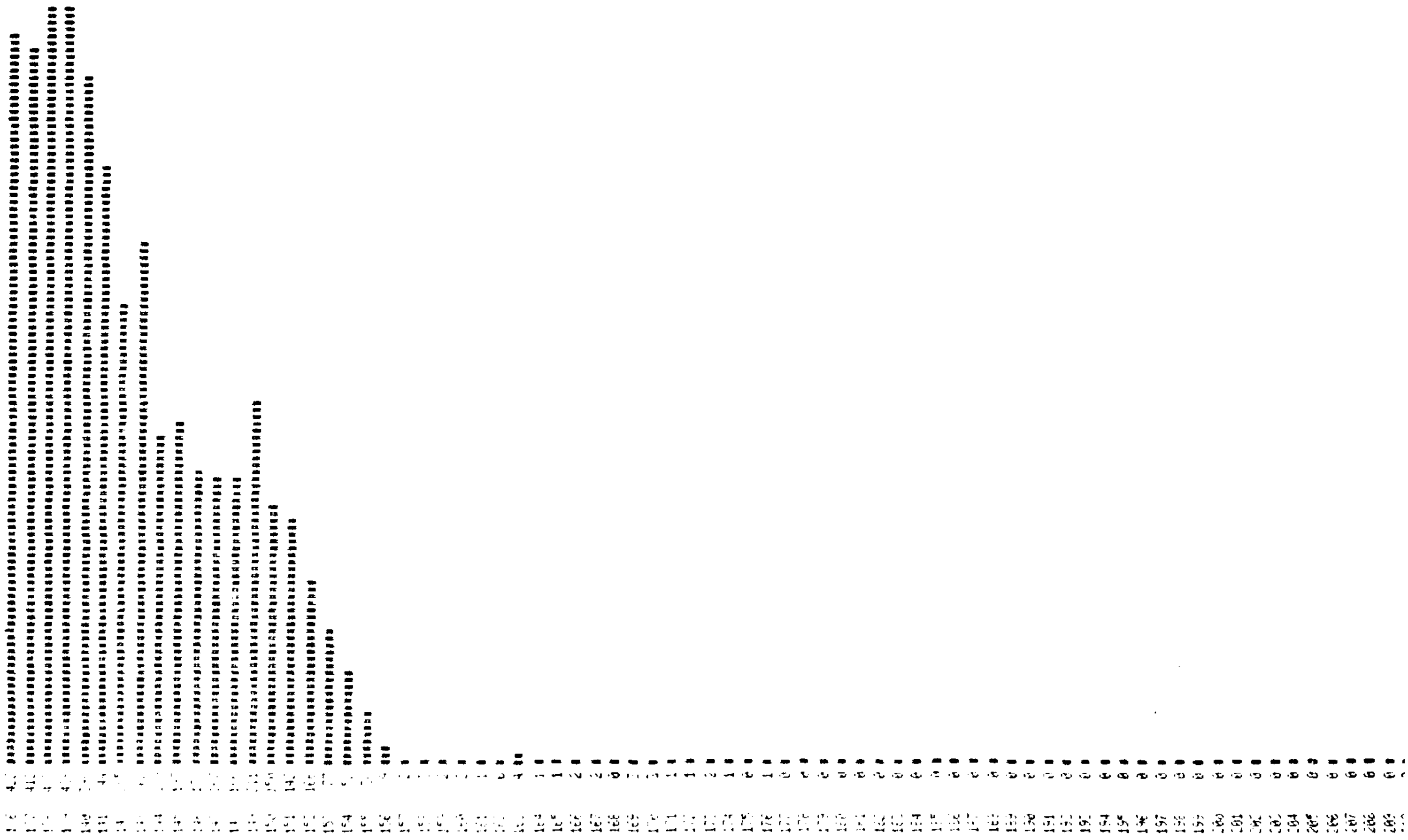
(H) Board crazing

(I) Legibility of screened, printed or etched marking

(J) Plating modulation in holes

(K) Holes with voids

(L) Bridged circuit lines



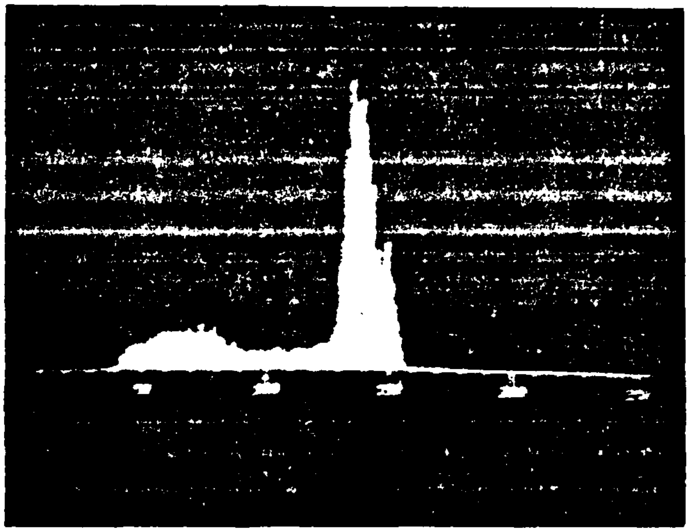


FIGURE 7 CRT DISPLAY BY HISTOGRAM

```
PROBABLE ERROR ON BOARD
  FILENAME IS   XXX3        LOWER THRESHOLD IS    15  UPPER THRESHOLD IS    95  TOTAL POINTS TO BE PLOTTED IS    2327
```

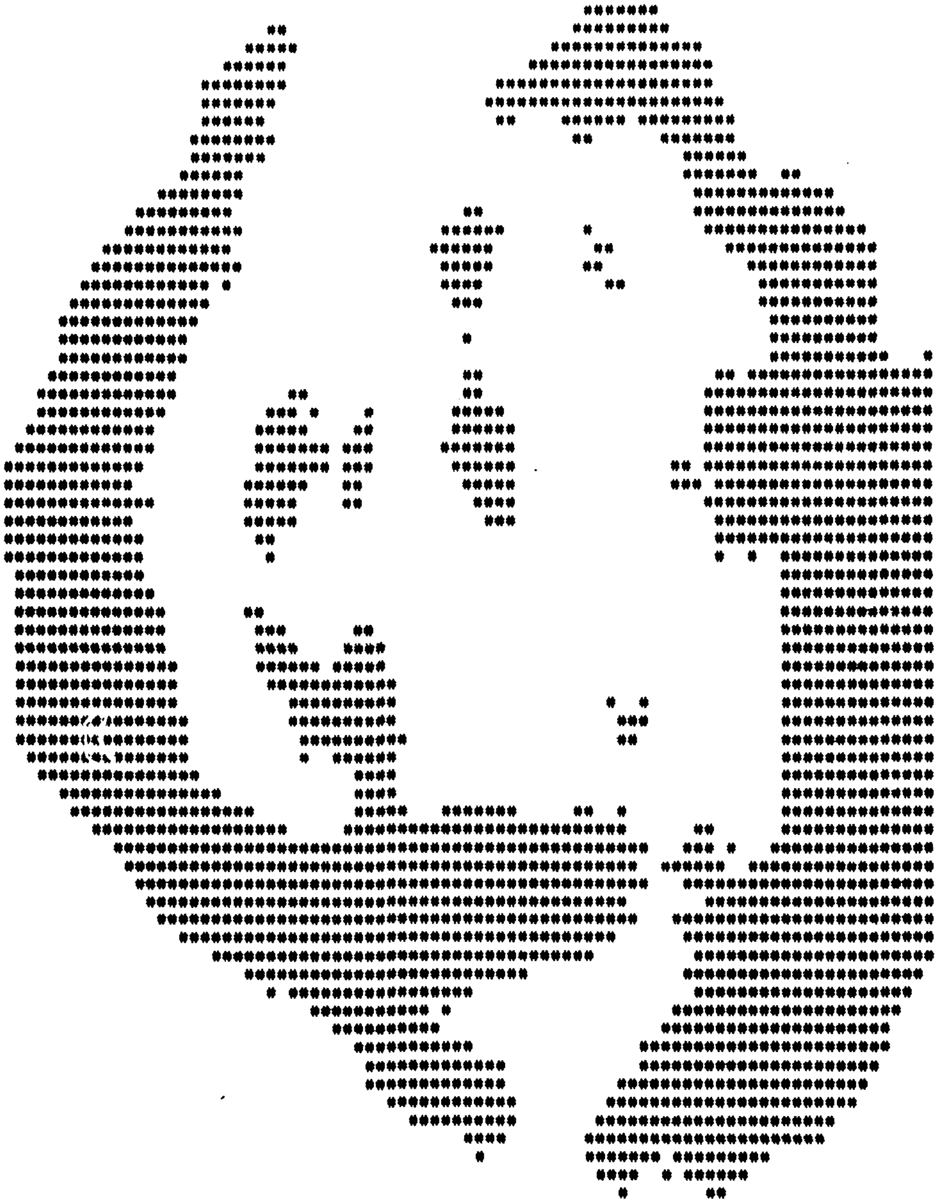


FIGURE 8   EXAMPLE OF MEASLING

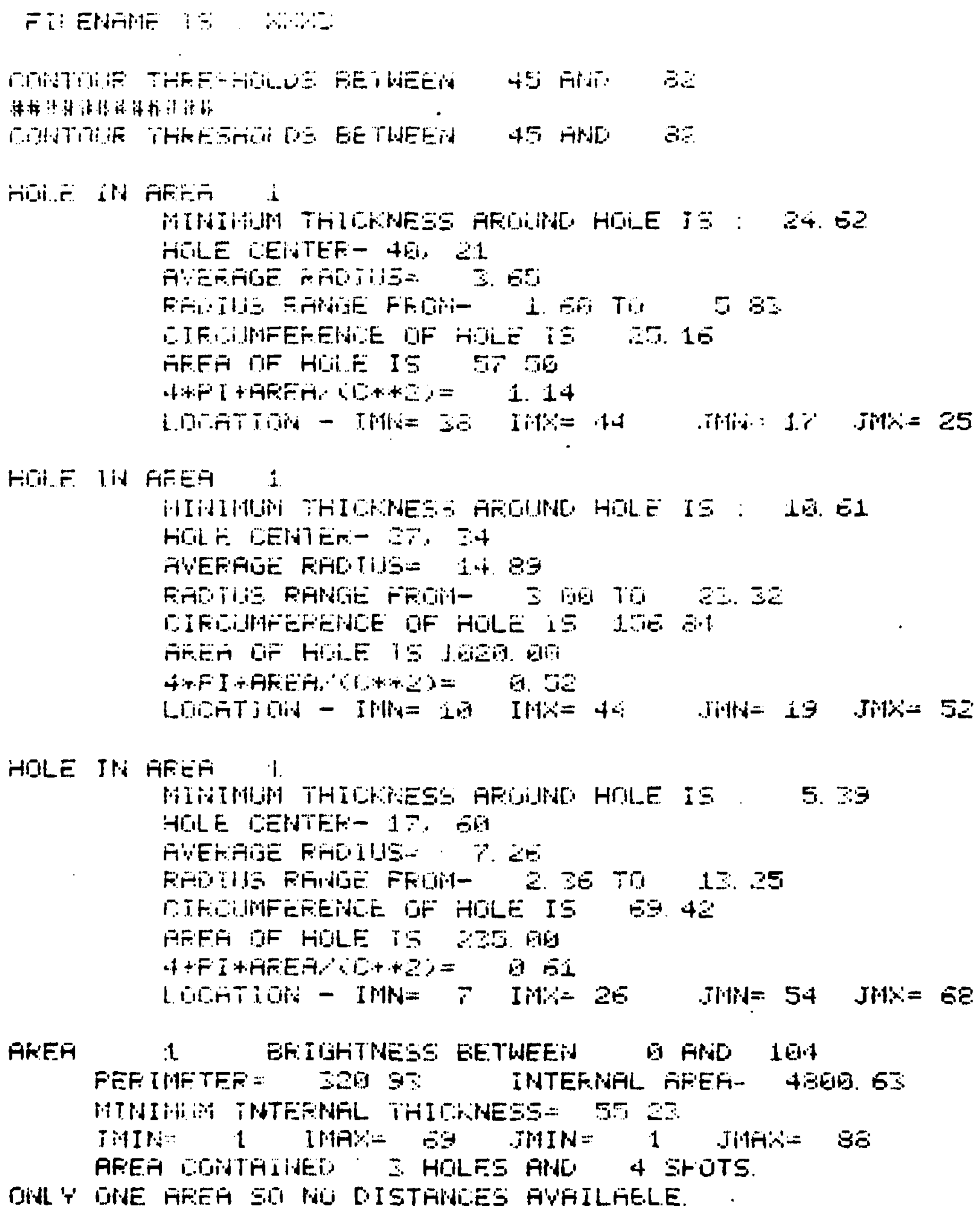
The type of system faults that are common on printed circuit cards are as follows:

1. Base Electrical
   - A. Delemination
   - B. Measling
2. Component Alignment
3. Conductor Flaw
4. Lifted Pad
5. Scratches
6. Surface Quality
7. Hole Registration
8. Legibility

The system can readily spot board elamination or measling but has difficulty in differentiating between the two. Component alignment checks have to be done on a looser tolerance. Conductor flaws on a bare board are the most consistently recognized of all. An example of measling is shown in Figure 8 with characteristics print on Figure 9.

Scratches or poor surface quality can be detected by change in the number of detected points using DOTMAT or PRTMAT. The use of PCCARD can then be used to isolate and measure the problem areas. This comparison technique requires that the light be maintained constant during a series of board to minimize the number of false alarms.

Legibility of letters and numbers can be examined by use of PCCARD to determine the minimum width on each letter or number and displaying the features on the Tektronix terminal. A scan of letters, in this case a logo, is shown on Figure 10.



```
 FILENAME IS  XXXX

CONTOUR THRE-HOLDS BETWEEN    45 AND    82
############
CONTOUR THRESHOLDS BETWEEN    45 AND    82

HOLE IN AREA    1
       MINIMUM THICKNESS AROUND HOLE IS :   24.62
       HOLE CENTER- 40, 21
       AVERAGE RADIUS=    3.65
       RADIUS RANGE FROM-    1.60 TO     5.83
       CIRCUMFERENCE OF HOLE IS    25.16
       AREA OF HOLE IS   57.50
       4*PI*AREA/(C**2)=    1.14
       LOCATION - IMN= 38  IMX= 44     JMN= 17  JMX= 25

HOLE IN AREA    1
       MINIMUM THICKNESS AROUND HOLE IS :   10.61
       HOLE CENTER- 27, 34
       AVERAGE RADIUS=  14.89
       RADIUS RANGE FROM-    3.00 TO    21.32
       CIRCUMFERENCE OF HOLE IS  156.84
       AREA OF HOLE IS 1020.00
       4*PI*AREA/(C**2)=    0.52
       LOCATION - IMN= 10  IMX= 44     JMN= 19  JMX= 52

HOLE IN AREA    1
       MINIMUM THICKNESS AROUND HOLE IS      5.39
       HOLE CENTER- 17, 60
       AVERAGE RADIUS=   7.26
       RADIUS RANGE FROM-    2.36 TO    13.25
       CIRCUMFERENCE OF HOLE IS    69.42
       AREA OF HOLE IS  235.00
       4*PI*AREA/(C**2)=    0.61
       LOCATION - IMN=  7  IMX= 26     JMN= 54  JMX= 62

AREA       1      BRIGHTNESS BETWEEN     0 AND   104
    PERIMETER=    320.93      INTERNAL AREA-   4800.63
    MINIMUM INTERNAL THICKNESS=  55.23
    IMIN=    1    IMAX=   69    JMIN=    1    JMAX=   88
    AREA CONTAINED    3 HOLES AND    4 SPOTS.
ONLY ONE AREA SO NO DISTANCES AVAILABLE.
```

FIGURE 9   PCCARD OUTPUT FOR MEASLING EXAMPLE

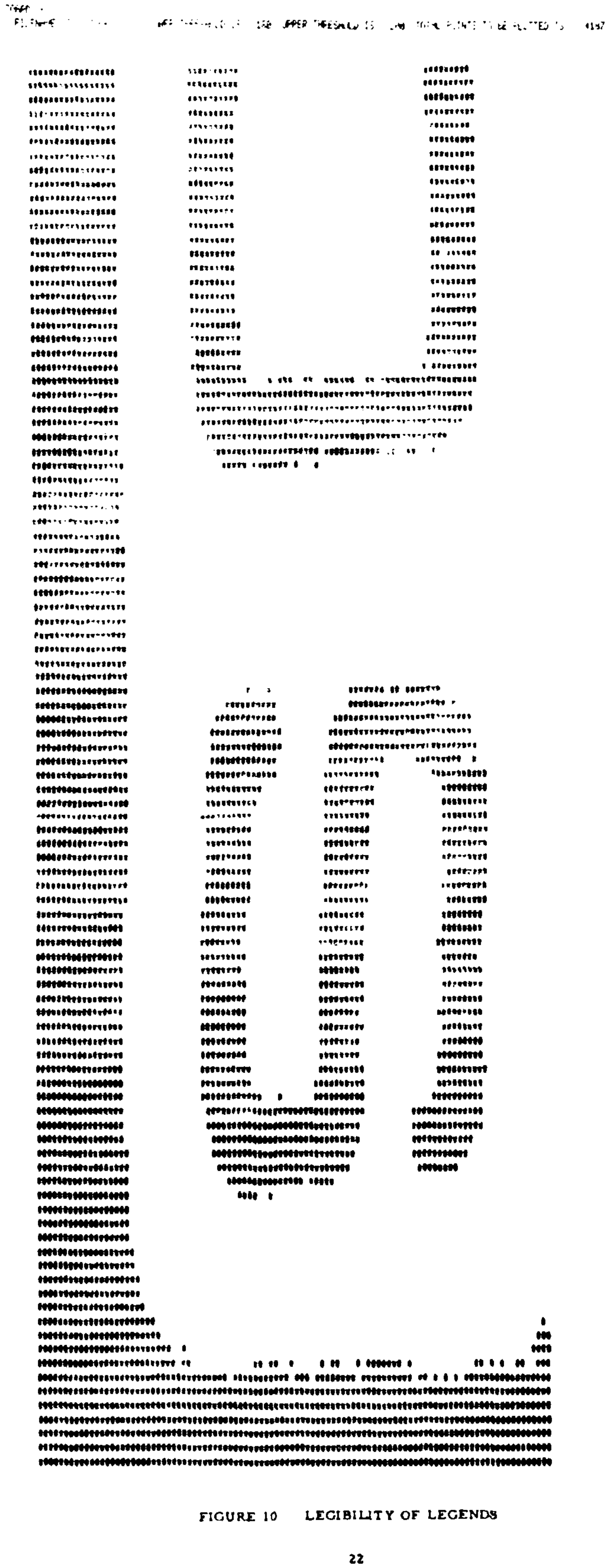


FIGURE 10 LEGIBILITY OF LEGENDS

Other detectable defects are shown in other figures included. Figures 11 and 12 illustrate a concentricity defect. Figures 13 and 14 illustrate an example of nodule defect. Figures 15 and 16 depict the same board without a nodule. Figure 18 displays the results from PRTMAT of a scratched board. Figure 18 is a display of the output of a routine named CALGRID which provides a method of calibration of absolute distances. Figures 19 A and B present the image of a broken land and Figures 20 A and B present an image of the same board without any broken land. Figure 21 illustrates the general system configuration.

```
PROBABLE ERROR ON BOARD
  FILENAME IS   XXX1         LOWER THRESHOLD IS      4  UPPER THRESHOLD IS   117  TOTAL POINTS TO BE PLOTTED IS    3196
```

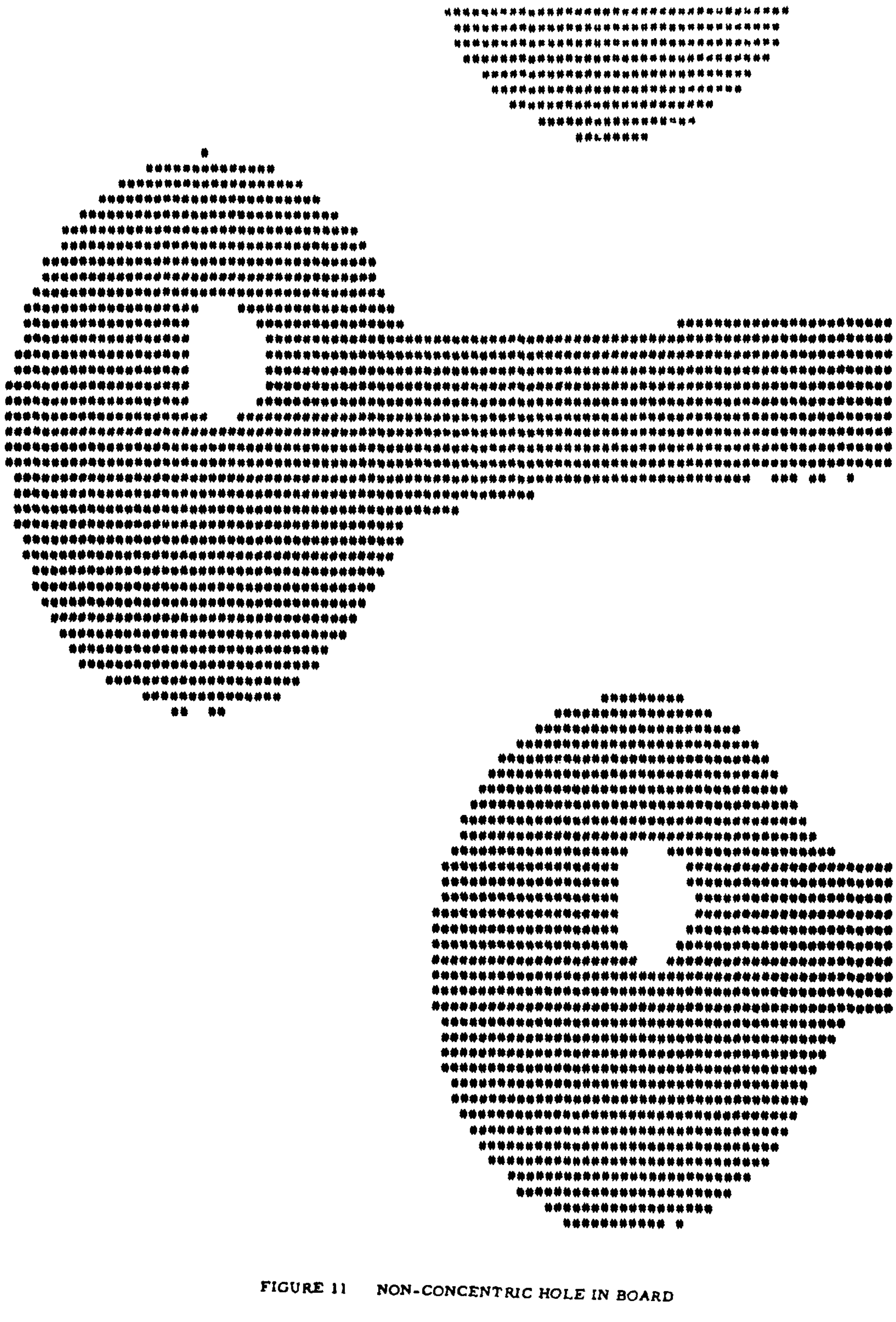


FIGURE 11   NON-CONCENTRIC HOLE IN BOARD

```
FILENAME IS   XXX1

CONTOUR THRESHOLDS BETWEEN    37 AND  100

HOLE IN AREA    1
          MINIMUM THICKNESS AROUND HOLE IS .     9. 62
          HOLE CENTER- 23, 71
          AVERAGE RADIUS=    5. 30
          RADIUS RANGE FROM-    3. 91 TO     6. 93
          CIRCUMFERENCE OF HOLE IS   34. 01
          AREA OF HOLE IS  103. 75
          4*PI*AREA/(C**2)=    1. 13
          LOCATION - IMN= 19  IMX= 28    JMN= 67  JMX= 76

AREA :    1      BRIGHTNESS BETWEEN     0 AND  106
     PERIMETER=   244. 19       INTERNAL AREA=  1846. 88
     MINIMUM INTERNAL THICKNESS=   8. 75
     IMIN=  11    IMAX=  45    JMIN=   1    JMAX=  94
     AREA CONTAINED   1 HOLES AND   0 SPOTS.

HOLE IN AREA    2
          MINIMUM THICKNESS AROUND HOLE IS :   10. 00
          HOLE CENTER- 58, 25
          AVERAGE RADIUS=    5. 25
          RADIUS RANGE FROM-    3. 91 TO     6. 93
          CIRCUMFERENCE OF HOLE IS   34. 01
          AREA OF HOLE IS  103. 75
          4*PI*AREA/(C**2)=    1. 13
          LOCATION - IMN= 54  IMX= 63    JMN= 21  JMX= 30

AREA :    2      BRIGHTNESS BETWEEN     0 AND  106
     PERIMETER=   146. 73       INTERNAL AREA=  1318. 75
     MINIMUM INTERNAL THICKNESS=  10. 00
     IMIN=  46    IMAX=  78    JMIN=   1    JMAX=  48
     AREA CONTAINED   1 HOLES AND   0 SPOTS.
AREA THICKNESS INDETERMINATE

AREA :    3      BRIGHTNESS BETWEEN     0 AND  106
     PERIMETER=    79. 06       INTERNAL AREA=   225. 63
     MINIMUM INTERNAL THICKNESS= *20000
     IMIN=   1    IMAX=   8    JMIN=  12    JMAX=  47
     AREA CONTAINED   0 HOLES AND   0 SPOTS.
THERE WERE   3 AREAS LOCATED.
MINIMUM DISTANCE FROM AREA    1 TO   2 IS    19. 17
MINIMUM DISTANCE FROM AREA    1 TO   3 IS    17. 50
 17. 50    22   24       8   24
 19. 17    31   38      46   34
 17. 50    22   24       8   24
```

FIGURE 12A   DATA FOR BOARD IN FIGURE 11

```
BOARD OK
  FILENAME IS   [illegible]          LOWER THRESHOLD IS      4  UPPER THRESHOLD IS    117
```

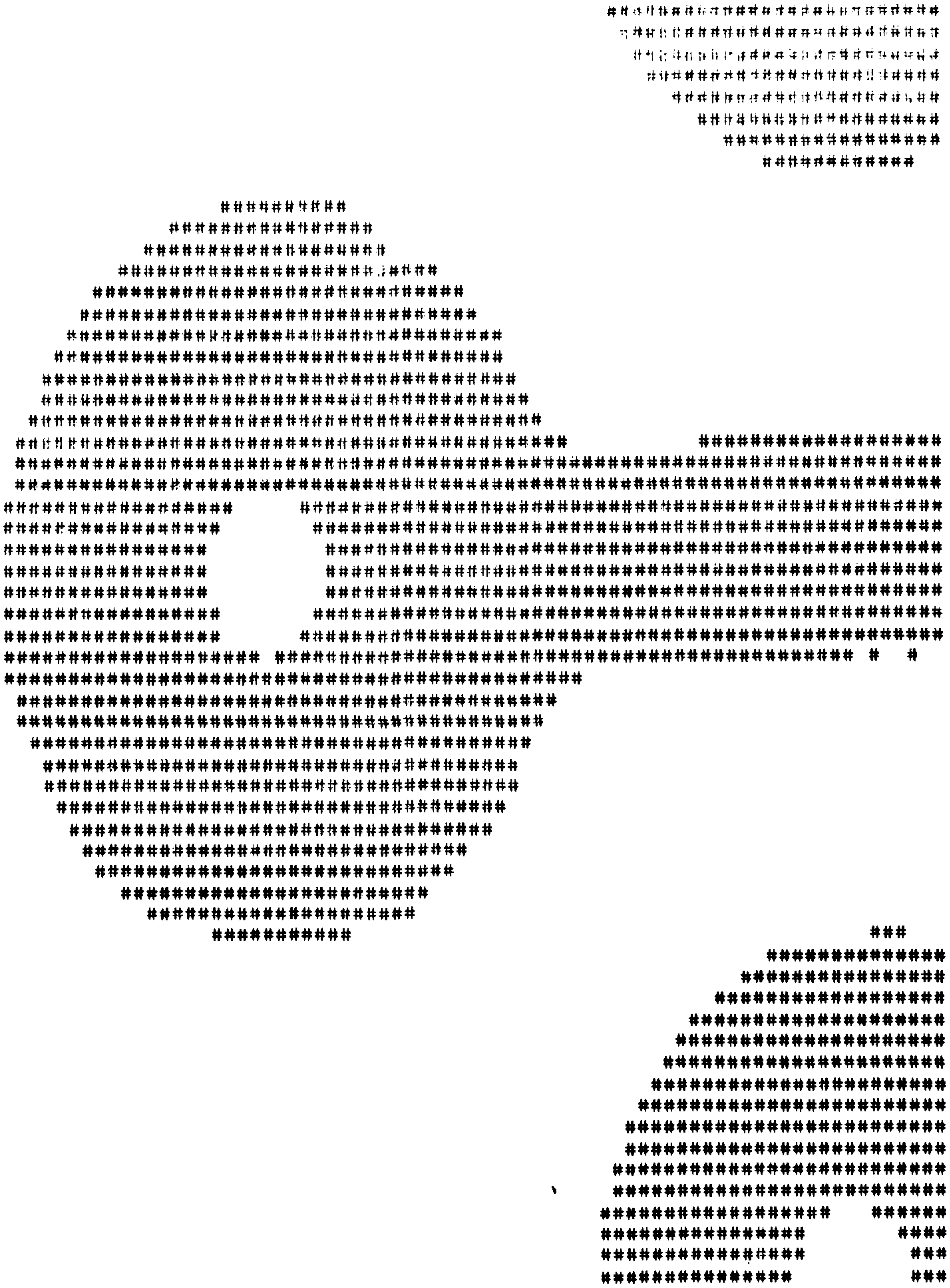


FIGURE 12 B CONCENTRIC HOLE

```
PROBABLE ERROR ON BOARD
  FILENAME IS   PCNOD        LOWER THRESHOLD IS    60  UPPER THRESHOLD IS   120  TOTAL POINTS TO BE PLOTTED IS    1805
```

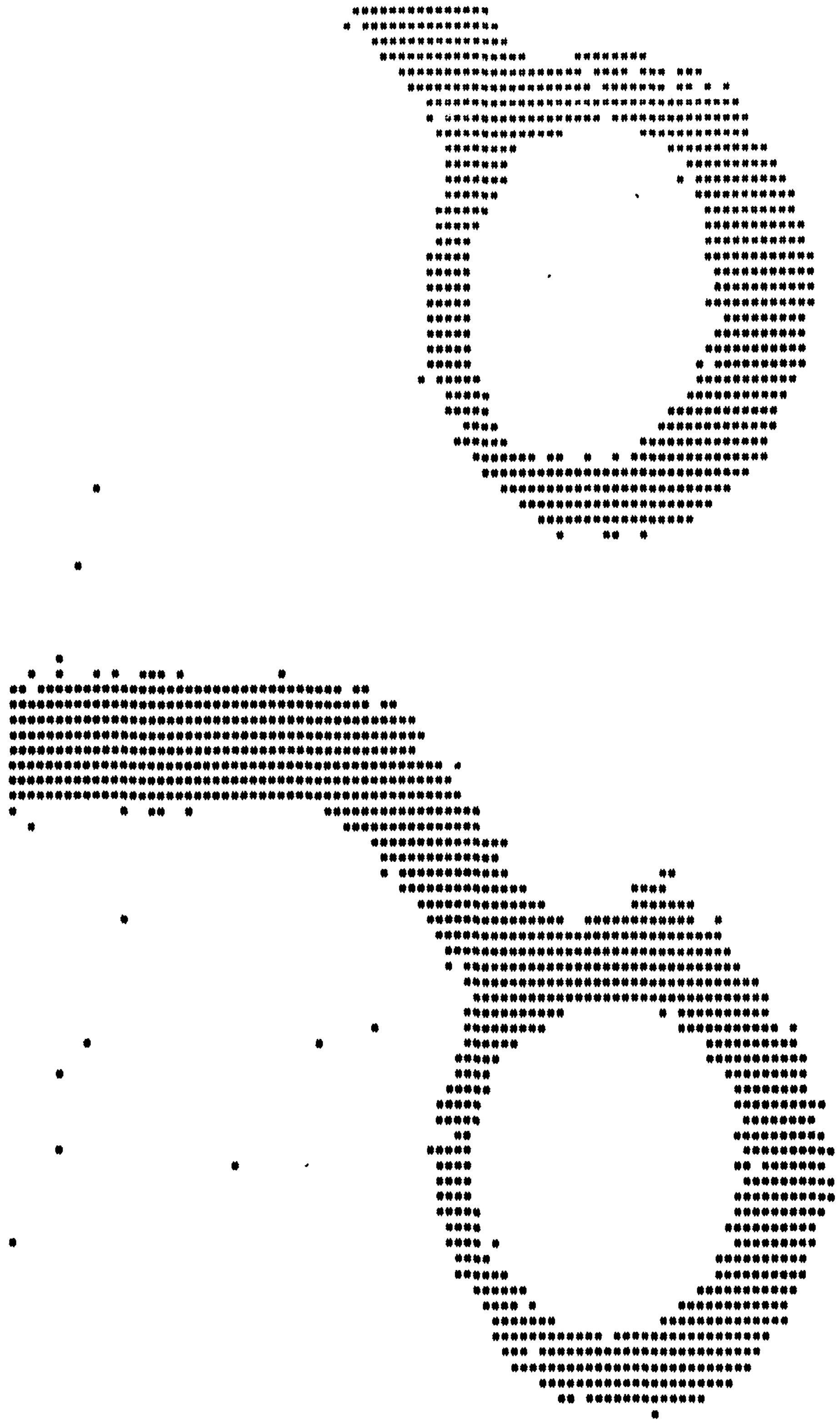
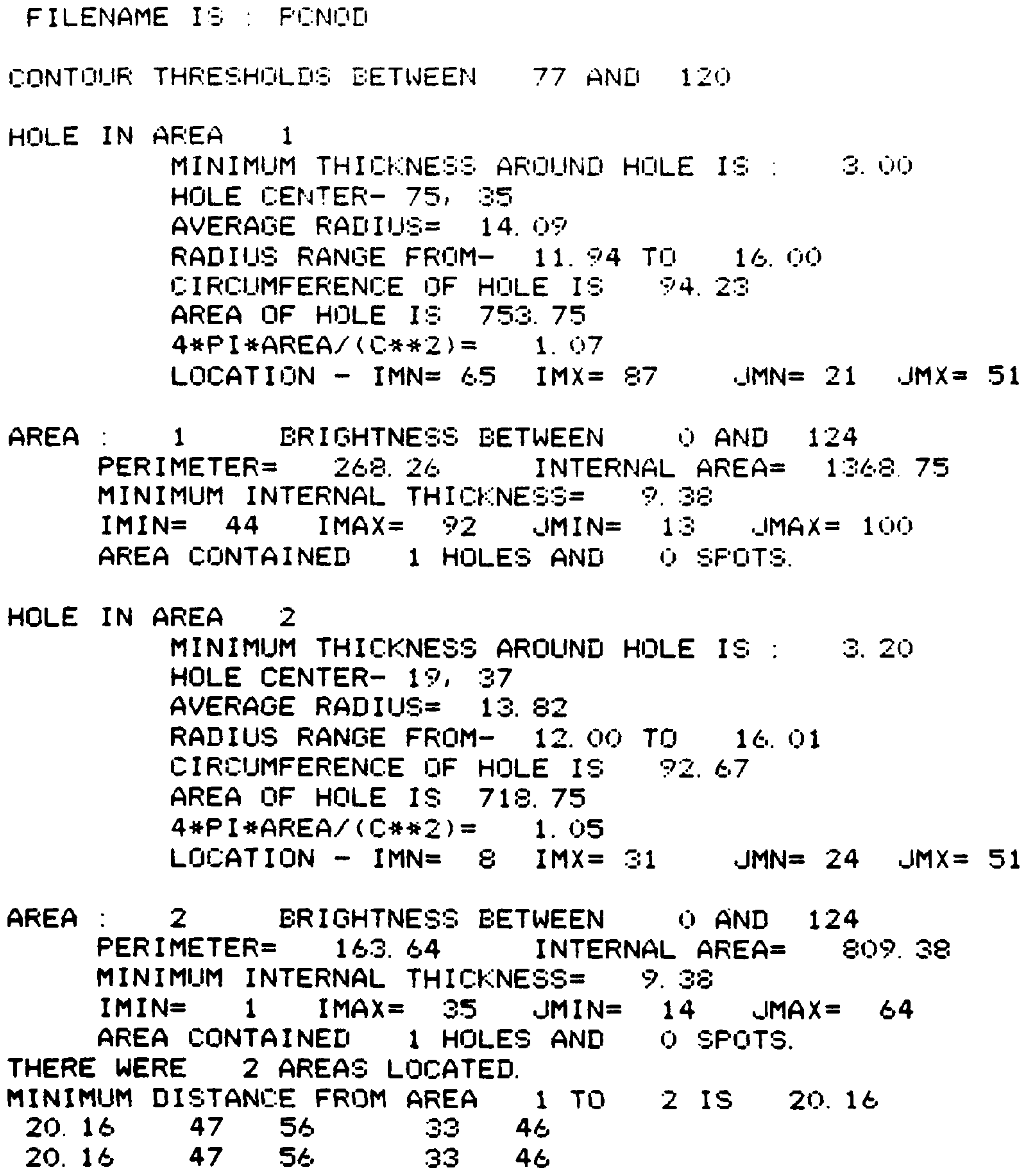


FIGURE 13   EXAMPLE OF NODULE



```
 FILENAME IS : PCNOD

CONTOUR THRESHOLDS BETWEEN   77 AND  120

HOLE IN AREA   1
         MINIMUM THICKNESS AROUND HOLE IS :    3.00
         HOLE CENTER- 75, 35
         AVERAGE RADIUS=  14.09
         RADIUS RANGE FROM-  11.94 TO   16.00
         CIRCUMFERENCE OF HOLE IS   94.23
         AREA OF HOLE IS  753.75
         4*PI*AREA/(C**2)=   1.07
         LOCATION - IMN= 65  IMX= 87    JMN= 21  JMX= 51

AREA :    1      BRIGHTNESS BETWEEN    0 AND  124
     PERIMETER=   268.26      INTERNAL AREA=  1368.75
     MINIMUM INTERNAL THICKNESS=   9.38
     IMIN=  44   IMAX=  92   JMIN=  13   JMAX= 100
     AREA CONTAINED   1 HOLES AND   0 SPOTS.

HOLE IN AREA   2
         MINIMUM THICKNESS AROUND HOLE IS :    3.20
         HOLE CENTER- 19, 37
         AVERAGE RADIUS=  13.82
         RADIUS RANGE FROM-  12.00 TO   16.01
         CIRCUMFERENCE OF HOLE IS   92.67
         AREA OF HOLE IS  718.75
         4*PI*AREA/(C**2)=   1.05
         LOCATION - IMN=  8  IMX= 31    JMN= 24  JMX= 51

AREA :    2      BRIGHTNESS BETWEEN    0 AND  124
     PERIMETER=   163.64      INTERNAL AREA=   809.38
     MINIMUM INTERNAL THICKNESS=   9.38
     IMIN=   1   IMAX=  35   JMIN=  14   JMAX=  64
     AREA CONTAINED   1 HOLES AND   0 SPOTS.
THERE WERE   2 AREAS LOCATED.
MINIMUM DISTANCE FROM AREA   1 TO   2 IS   20.16
 20.16    47   56      33   46
 20.16    47   56      33   46
```

FIGURE 14   DATA FOR BOARD IN FIGURE 13

```
FILENAME IS   [illegible]      LOWER THRESHOLD IS    68  UPPER THRESHOLD IS   114  TOTAL POINTS TO BE PLOTTED IS    1722
```

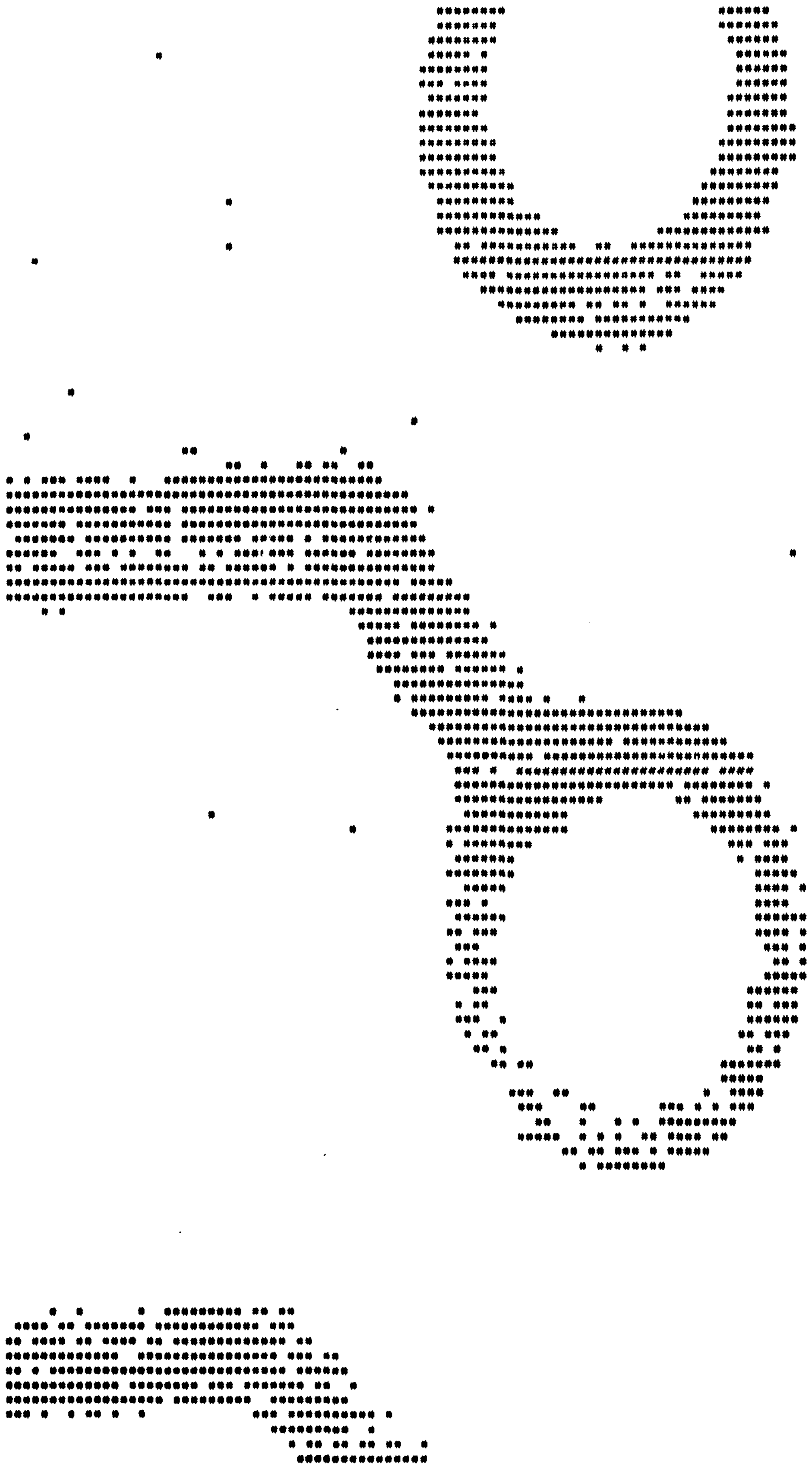
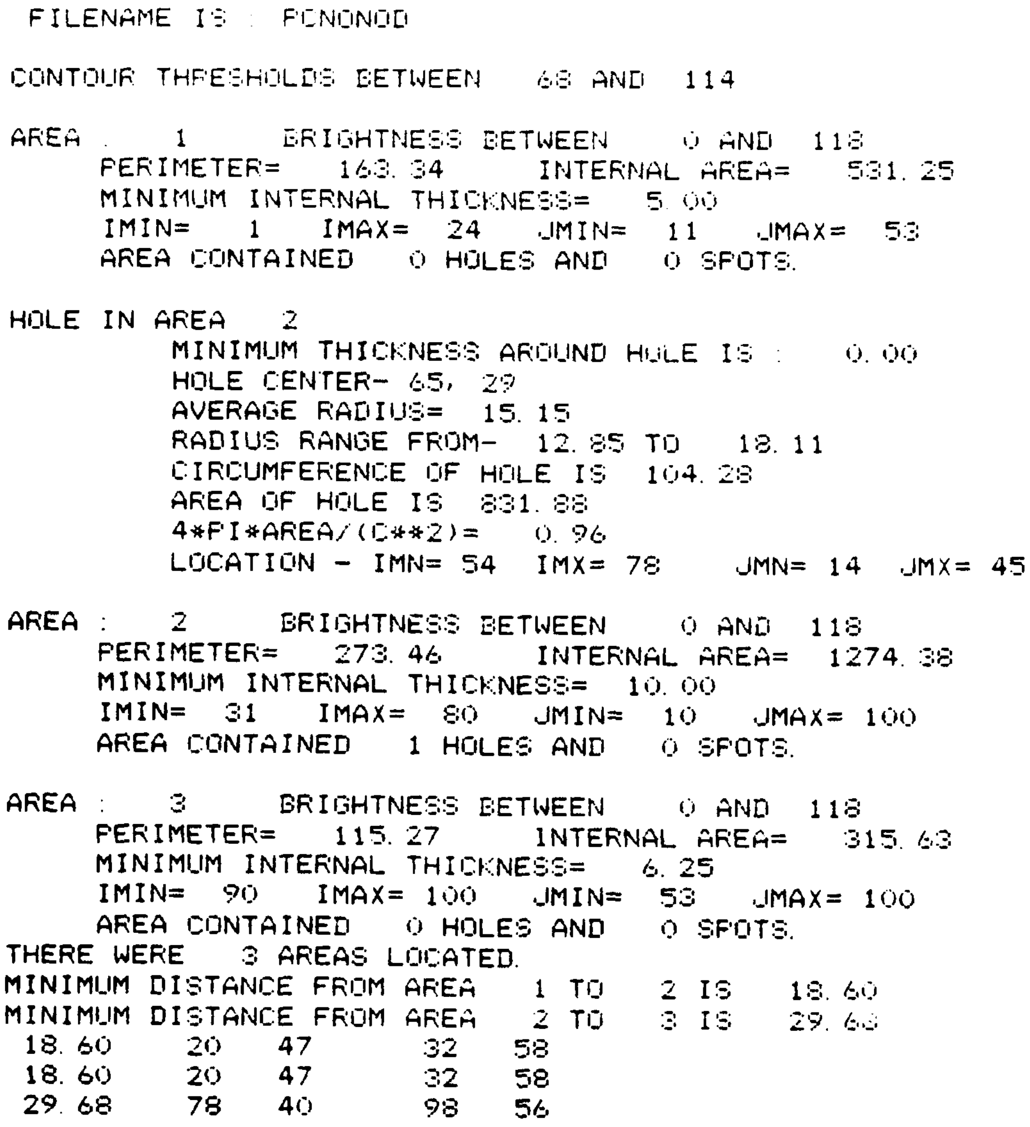


FIGURE 15   BOARD WITHOUT A NODULE



```
 FILENAME IS : PCNONOD

CONTOUR THRESHOLDS BETWEEN   68 AND  114

AREA :    1      BRIGHTNESS BETWEEN     0 AND  118
     PERIMETER=   163. 34     INTERNAL AREA=   531. 25
     MINIMUM INTERNAL THICKNESS=   5. 00
     IMIN=   1   IMAX=  24   JMIN=  11   JMAX=  53
     AREA CONTAINED    0 HOLES AND    0 SPOTS.

HOLE IN AREA   2
         MINIMUM THICKNESS AROUND HOLE IS :    0. 00
         HOLE CENTER- 65, 29
         AVERAGE RADIUS=  15. 15
         RADIUS RANGE FROM-  12. 85 TO   18. 11
         CIRCUMFERENCE OF HOLE IS  104. 28
         AREA OF HOLE IS  831. 88
         4*PI*AREA/(C**2)=   0. 96
         LOCATION - IMN= 54  IMX= 78    JMN= 14  JMX= 45

AREA :    2      BRIGHTNESS BETWEEN     0 AND  118
     PERIMETER=   273. 46     INTERNAL AREA=  1274. 38
     MINIMUM INTERNAL THICKNESS=  10. 00
     IMIN=  31   IMAX=  80   JMIN=  10   JMAX= 100
     AREA CONTAINED    1 HOLES AND    0 SPOTS.

AREA :    3      BRIGHTNESS BETWEEN     0 AND  118
     PERIMETER=   115. 27     INTERNAL AREA=   315. 63
     MINIMUM INTERNAL THICKNESS=   6. 25
     IMIN=  90   IMAX= 100   JMIN=  53   JMAX= 100
     AREA CONTAINED    0 HOLES AND    0 SPOTS.
THERE WERE   3 AREAS LOCATED.
MINIMUM DISTANCE FROM AREA   1 TO   2 IS   18. 60
MINIMUM DISTANCE FROM AREA   2 TO   3 IS   29. 68
 18. 60     20    47        32    58
 18. 60     20    47        32    58
 29. 68     78    40        98    56
```

FIGURE 16   DATA FOR BOARD IN FIGURE 15

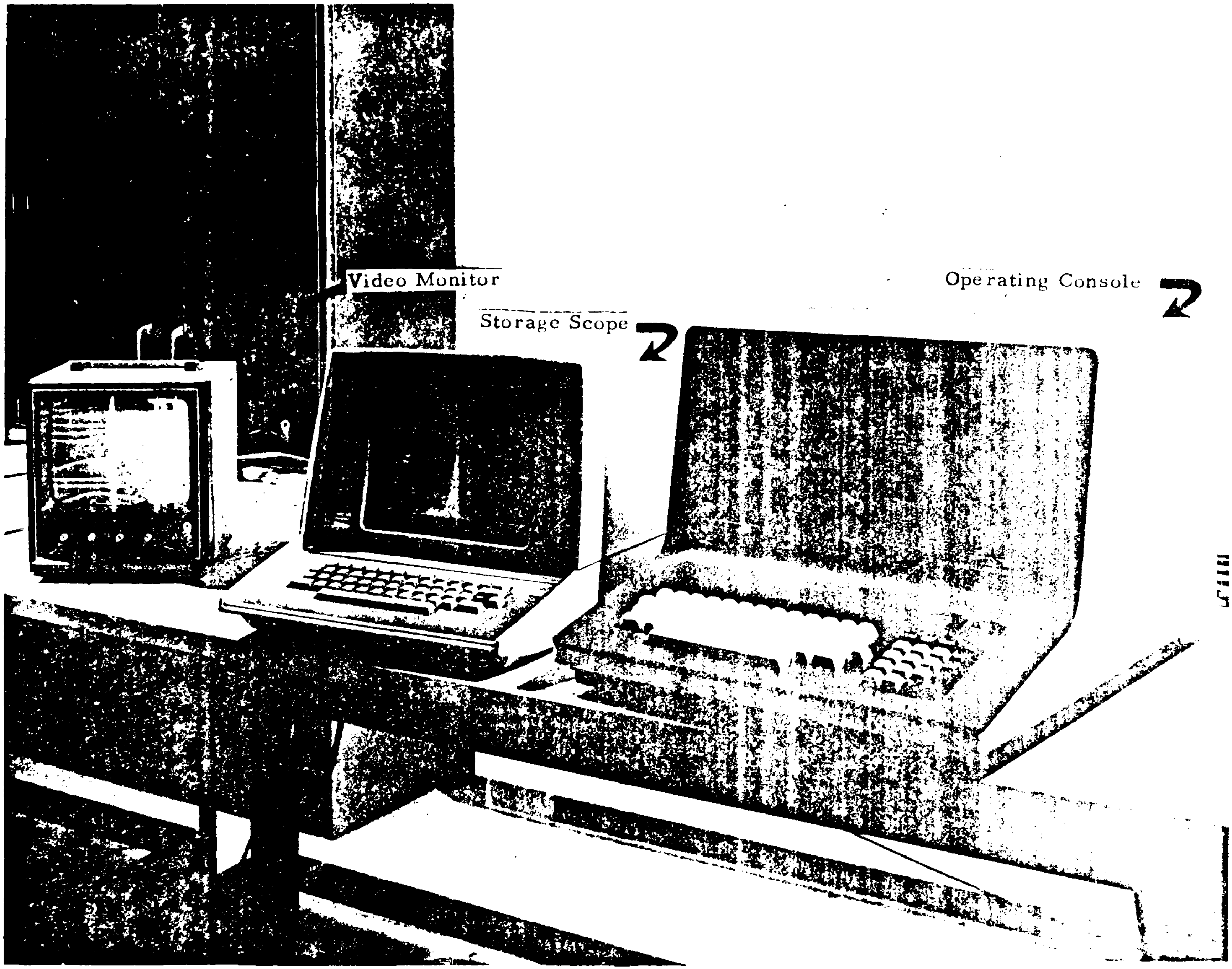


Figure A-1 Data General Terminal, Storage Scope & Monitor

APPENDIX A

ELEMENTS OF THE SYSTEM

more rapid inspection and would reduce the amount of memory required. The hardware would not access thru except when the desired attribution in the data are found.

5. A relatively small computer, possibly a microprocessor. To produce the necessary report to go with each board. With the hardware preprocessing, a relatively small amount of computation would be necessary.

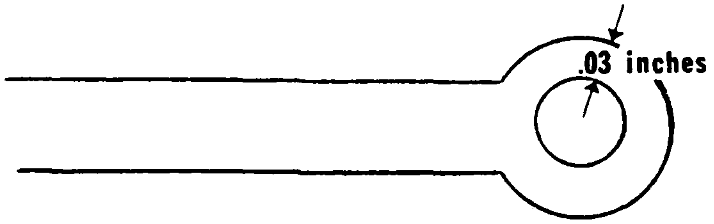
holes can be detected accurately enough. The problem then reduces to finding a device with a large number of resolution elements, and a fast scan rate. The best choice for this is probably one of the solid state line arrays. This type of array could be combined with a board moving mechanism (like a conveyor belt) to give a rapid high resolution scan. The software would need to be changed to accumulate the features of the areas in real time so that storage of the complete image would not be necessary. The storage of complete images must be eliminated if high resolution scans of moderate sized boards are to be accomplished. In order to get a greater processing speed, it may be necessary to do a portion of the computation in hardware.

Additional work can be done in the thresholding. The present algorithms work well if the image illumination is flat, but if there is shading of the field, then the segmentation is poorly done. Some image processing systems have eliminated this shading problem with hardware filters. This would certainly be a plausable approach.

The final recommendations for inspection of drilled and etched boards would be to use:

1. A solid state line array for rapid scanning.
2. A continuously moving PC board ideal for production.
3. Hardware Dynamic Thresholding - In order to do better scene segmentation, a thresholding algorithm based on rate of change of contrast will probably need to be implemented in hardware. The task of thresholding and scene segmentation is crucial to all of the image processing and should receive maximum attention.
4. Hardware Processing - Hardware Processing for features such as area, perimeter, minimum thickness, etc., would permit

While the current system is a very excellent analytical system, the routines would of course benefit from more on-line refinement of the diagnostic and detection methods employed. This would provide a greater attractiveness for production line applications of the system. One outstanding ability of the system is searching for dimensional variations and for minimum distances. Lack of concentricity in hole locations is an example, vis-a-vis:



Using this ability an inspection could be made of a sample lot and the resulting distribution of dimensional variances could be used to characterize the quality control for that particular production run.

The next effort to be made in the complete board automatic inspection should probably be in the training portion. That is, a technique for storing the data on a "good" board in some compacted form should be developed. One suggested technique is to store the printout data for each portion of the board scanned. This is considerably more compact than storing image files and has the advantage that it is less sensitive to alignment. The particular portion of the printout data that would be useful is the number of area's, their perimeters, their areas, and the number of holes. The distances do not need to be saved since they are covered by specification.

A limiting factor on the speed of inspection is the scanning device.

The scanning resolution of approximately 0.004" needs to be maintained so that the widths can be resolved and the hole roundness on 0.020 inch di meter

The fact that the computer is tireless permits 100% inspection of boards to the spacing and line width specifications. It is also simple to 100% inspect for hole roundness.

The automated system can detect delamination and measling, if the defect is severe enough to be high enough contrast, and if the defect is in an area that causes the width of spacing to be affected. Conductor flaws can also be easily detected, if they affect line widths. Conductor flaws that appear as normal runs with breaks in them can be missed by the automated system unless the system uses a training board.

The defects on soldered boards were not studies using this system since the illumination scheme did not give satisfactory differentiation between soldered areas and board material. Therefore, defects such as lifted pads were not detected, nor were any of the defects that appear only on soldered boards.

Component alignment was not readily detected since the contrast on a component generally varies too greatly for the scene segmentation schemes which were used to operate satisfactorily.

Scratches, surface quality (within limits), and hole registration were readily detected.

Legibility of legends is a nebulous defect, but if the legends are designed to the same specifications in line width and spacing (a very reasonable approach!) then their inspection would be identical to that of the rest of the board.

IV. CONCLUSIONS & RECOMMENDATIONS

The system as configured has shown exceptional resolution and good sensitivity. The use of the automatic threshold setting feature proved to be less than ideal except for a given set of lighting conditions. To gain greater control over threshold settings, a manual entry option was added to the PCCARD Routine allowing entry of both lower and upper thresholds.

Sufficient repeatability was attained by use of mechanical indexing of the PC card.

One goal of this effort was to determine how much of the PC board inspection task could be done using the computer as an inspector, so that it could inspect a board without having seen a "training" board. The word on this particular technique was limited to dimensional analysis of a bare (unloaded-unsoldered) board. We found that the computer could readily inspect for minimum line widths and spacings, and that, at our high resolution, it could determine the quality of the holes in the etch. By measuring the ratio of the square of the perimeter to the area of the holes out of round holes or holes with modules could be detected. By pre-selecting line spacing and line width limitations and by pre-selecting the limitation on the "roundness" (P^2/A) criterion, fully automatic inspection is feasible for production rise. The scanning of a complete board and it's inspection for dimensional tolerances can be done by the computer with no need for any training. This is useful for areas of the board limited to normal point to point runs with round component holes. One interesting fact we discerned was that if the lettering in the legend area's of the board was not designed to the same line width and spacing as the rest of the board, it would be flagged as an out of specification area.

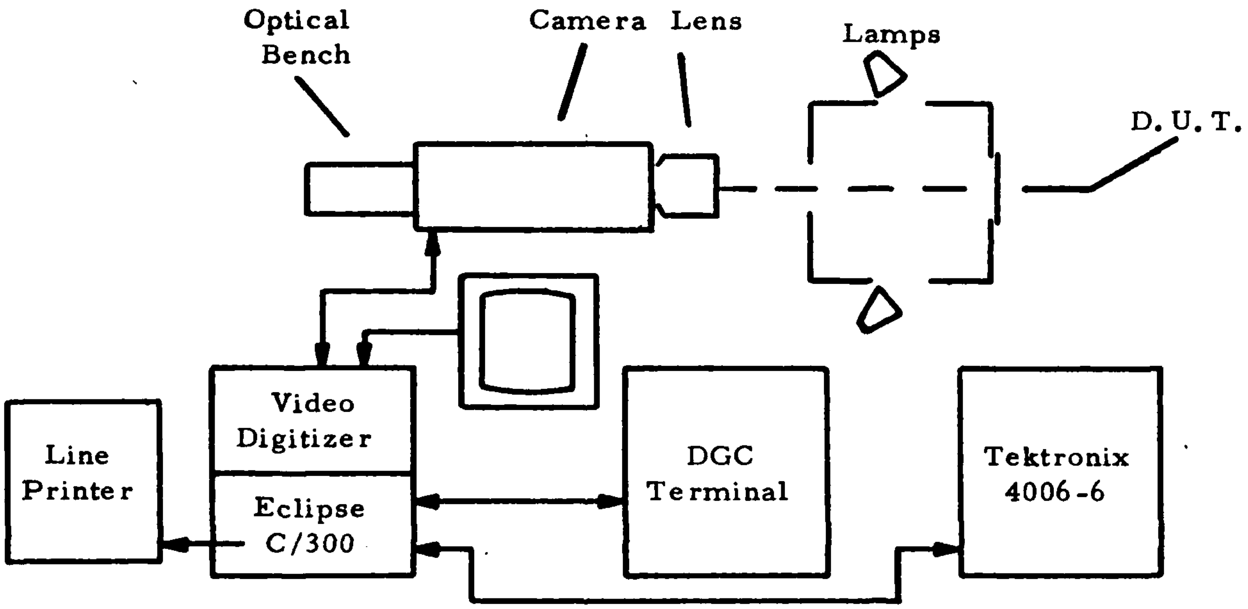
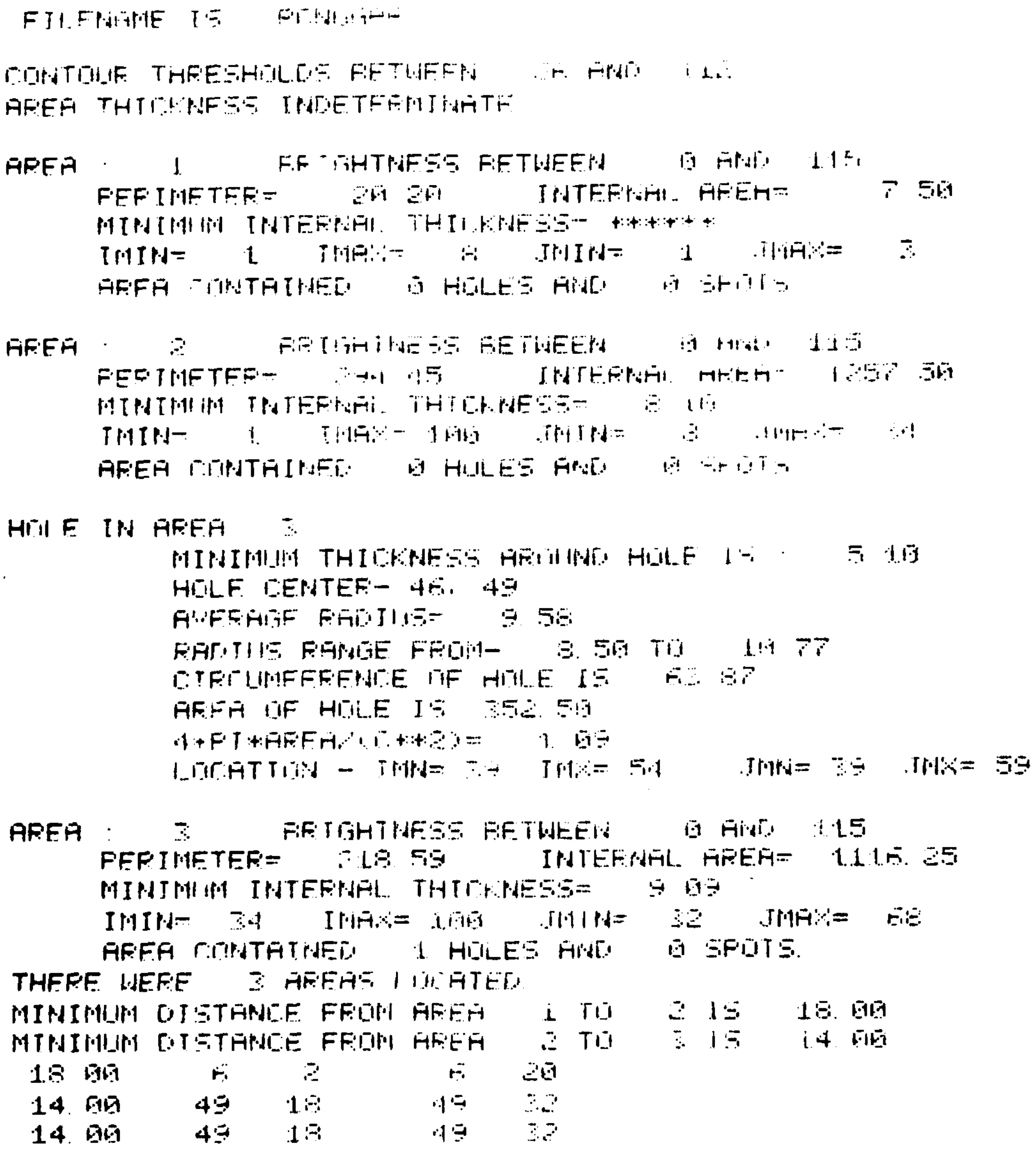


FIGURE 21 SYSTEM CONFIGURATION



```
 FILENAME IS   PCNGAA

CONTOUR THRESHOLDS BETWEEN   36 AND  115
AREA THICKNESS INDETERMINATE

AREA :    1      BRIGHTNESS BETWEEN     0 AND  115
     PERIMETER=     20.20      INTERNAL AREA=      7.50
     MINIMUM INTERNAL THICKNESS= ******
     IMIN=   1    IMAX=    8   JMIN=    1    JMAX=    3
     AREA CONTAINED    0 HOLES AND    0 SPOTS

AREA :    2      BRIGHTNESS BETWEEN     0 AND  115
     PERIMETER=   244.45       INTERNAL AREA=  1257.50
     MINIMUM INTERNAL THICKNESS=    8.10
     IMIN=   1    IMAX= 100    JMIN=    3    JMAX=   64
     AREA CONTAINED    0 HOLES AND    0 SPOTS

HOLE IN AREA    3
         MINIMUM THICKNESS AROUND HOLE IS      5.10
         HOLE CENTER- 46, 49
         AVERAGE RADIUS=    9.58
         RADIUS RANGE FROM-    8.50 TO    10.77
         CIRCUMFERENCE OF HOLE IS    63.87
         AREA OF HOLE IS  352.50
         4*PI*AREA/(C**2)=    1.09
         LOCATION - IMN= 34  IMX= 54     JMN= 39  JMX= 59

AREA :    3      BRIGHTNESS BETWEEN     0 AND  115
     PERIMETER=   218.59       INTERNAL AREA=  1116.25
     MINIMUM INTERNAL THICKNESS=    9.09
     IMIN=  34    IMAX= 100    JMIN=   32    JMAX=   68
     AREA CONTAINED    1 HOLES AND    0 SPOTS.
THERE WERE    3 AREAS LOCATED
MINIMUM DISTANCE FROM AREA    1 TO    2 IS    18.00
MINIMUM DISTANCE FROM AREA    2 TO    3 IS    14.00
 18.00      6     2         6    20
 14.00     49    18        49    32
 14.00     49    18        49    32
```

FIGURE 20-B   MEASUREMENTS FOR BOARD SHOWN IN FIGURE 20-A

```
BOARD OK
  FILENAME IS   PINDGAR     LOWER THRESHOLD IS      0  UPPER THRESHOLD IS   105
```

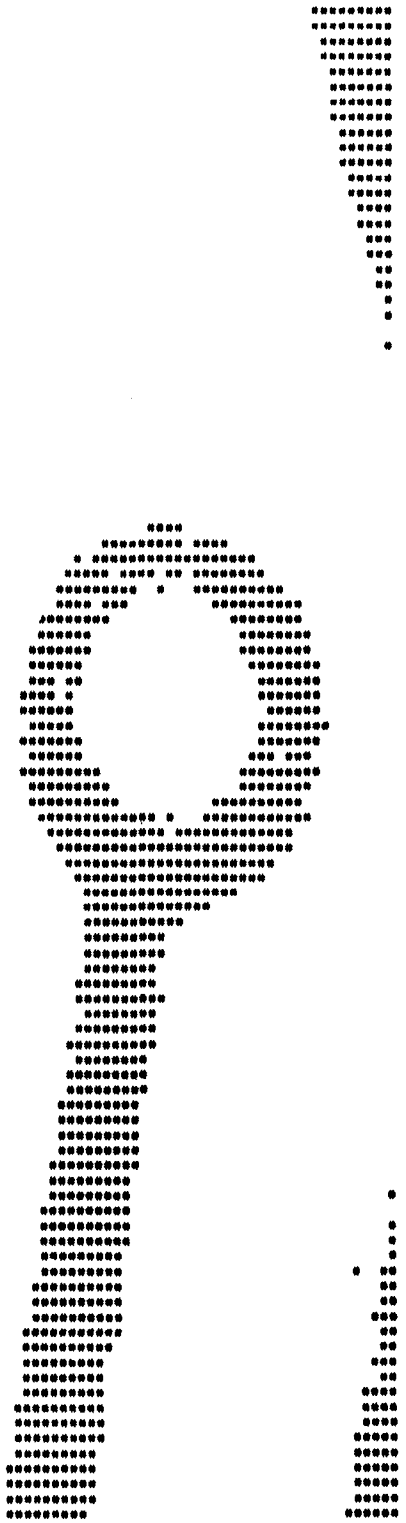


FIGURE 20-A   IMAGE OF GOOD BOARD (TYPE AS IN FIG. 19)

```
 FILENAME IS : PCGAP

CONTOUR THRESHOLDS BETWEEN   60 AND   99

AREA :    1      BRIGHTNESS BETWEEN     0 AND  102
     PERIMETER=   219. 19       INTERNAL AREA=   584. 38
     MINIMUM INTERNAL THICKNESS=   4. 19
     IMIN=   1    IMAX=  79   JMIN=   4   JMAX=  26
     AREA CONTAINED    0 HOLES AND    0 SPOTS.

AREA :    2      BRIGHTNESS BETWEEN     0 AND  102
     PERIMETER=     46. 51      INTERNAL AREA=     77. 50
     MINIMUM INTERNAL THICKNESS=   4. 00
     IMIN=  86    IMAX= 100   JMIN=  13   JMAX=  22
     AREA CONTAINED    0 HOLES AND    0 SPOTS.

HOLE IN AREA   3
         MINIMUM THICKNESS AROUND HOLE IS :     1. 00
         HOLE CENTER- 47, 40
         AVERAGE RADIUS=    9. 47
         RADIUS RANGE FROM-    7. 81 TO    11. 07
         CIRCUMFERENCE OF HOLE IS    63. 22
         AREA OF HOLE IS  347. 50
         4*PI*AREA/(C**2)=    1. 09
         LOCATION - IMN= 40  IMX= 55     JMN= 31  JMX= 51

AREA :    3      BRIGHTNESS BETWEEN     0 AND  102
     PERIMETER=    103. 08      INTERNAL AREA=   463. 13
     MINIMUM INTERNAL THICKNESS=  25. 71
     IMIN=  33    IMAX=  58   JMIN=  29   JMAX=  59
     AREA CONTAINED    1 HOLES AND    0 SPOTS.

AREA :    4      BRIGHTNESS BETWEEN     0 AND  102
     PERIMETER=     92. 96      INTERNAL AREA=   209 38
     MINIMUM INTERNAL THICKNESS=   4. 19
     IMIN=  68    IMAX= 100   JMIN=  48   JMAX=  59
     AREA CONTAINED    0 HOLES AND    0 SPOTS
AREA THICKNESS INDETERMINATE

AREA :    5      BRIGHTNESS BETWEEN     0 AND  102
     PERIMETER=   141. 66       INTERNAL AREA=     51. 88
     MINIMUM INTERNAL THICKNESS= *20000
     IMIN=  45    IMAX= 100   JMIN=  99   JMAX= 100
     AREA CONTAINED    0 HOLES AND    0 SPOTS.
THERE WERE    5 AREAS LOCATED.
MINIMUM DISTANCE FROM AREA    1 TO    2 IS     8. 75
MINIMUM DISTANCE FROM AREA    1 TO    3 IS    17. 67
MINIMUM DISTANCE FROM AREA    3 TO    4 IS    12. 50
MINIMUM DISTANCE FROM AREA    5 TO    4 IS   141. 42
MINIMUM DISTANCE FROM AREA    5 TO    1 IS   141. 42
MINIMUM DISTANCE FROM AREA    5 TO    2 IS   141. 42
MINIMUM DISTANCE FROM AREA    5 TO    3 IS   141. 42
   8. 75     79    13       86    13
   8. 75     79    13       86    13
  12. 50     58    50       68    50
  12. 50     58    50       68    50
 141. 42     58    50       68    50
```

FIGURE 19B   DATA FOR FIGURE 19A

```
[illegible]
[illegible]        L[illegible] THRESHOLD I[illegible]      [illegible] UPPER THRESHOLD IS    [illegible]  TOTAL POINTS TO BE PLOTTED IS     1158
```

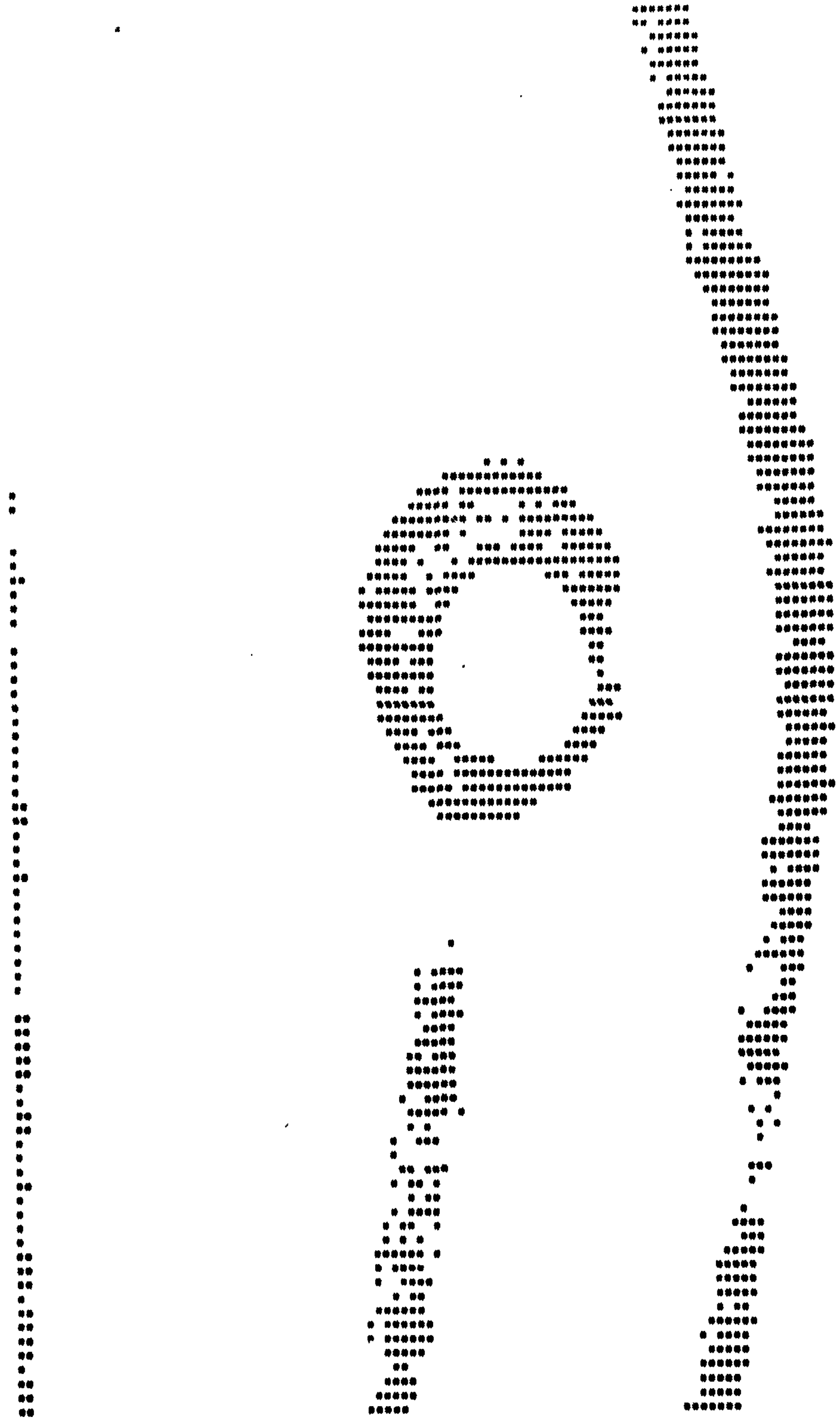


FIGURE 19A   BOARD WITH GAP

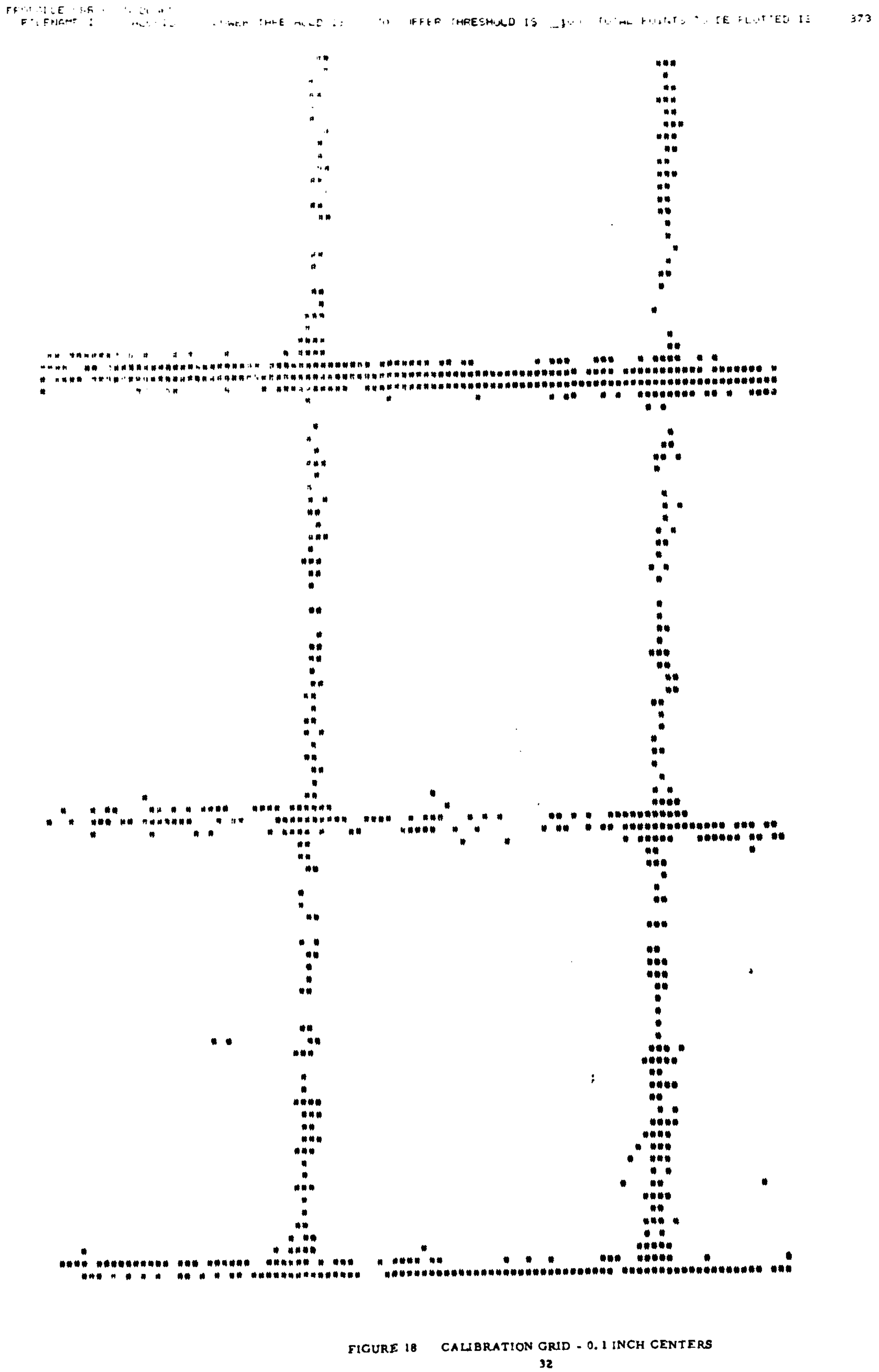


FIGURE 18 CALIBRATION GRID - 0.1 INCH CENTERS

```
PROBABLE ERROR IN BOARD
  FILENAME IS   [illegible]7        LOWER THRESHOLD IS    50  UPPER THRESHOLD IS   150  TOTAL POINTS TO BE PLOTTED IS    62[illegible]
```

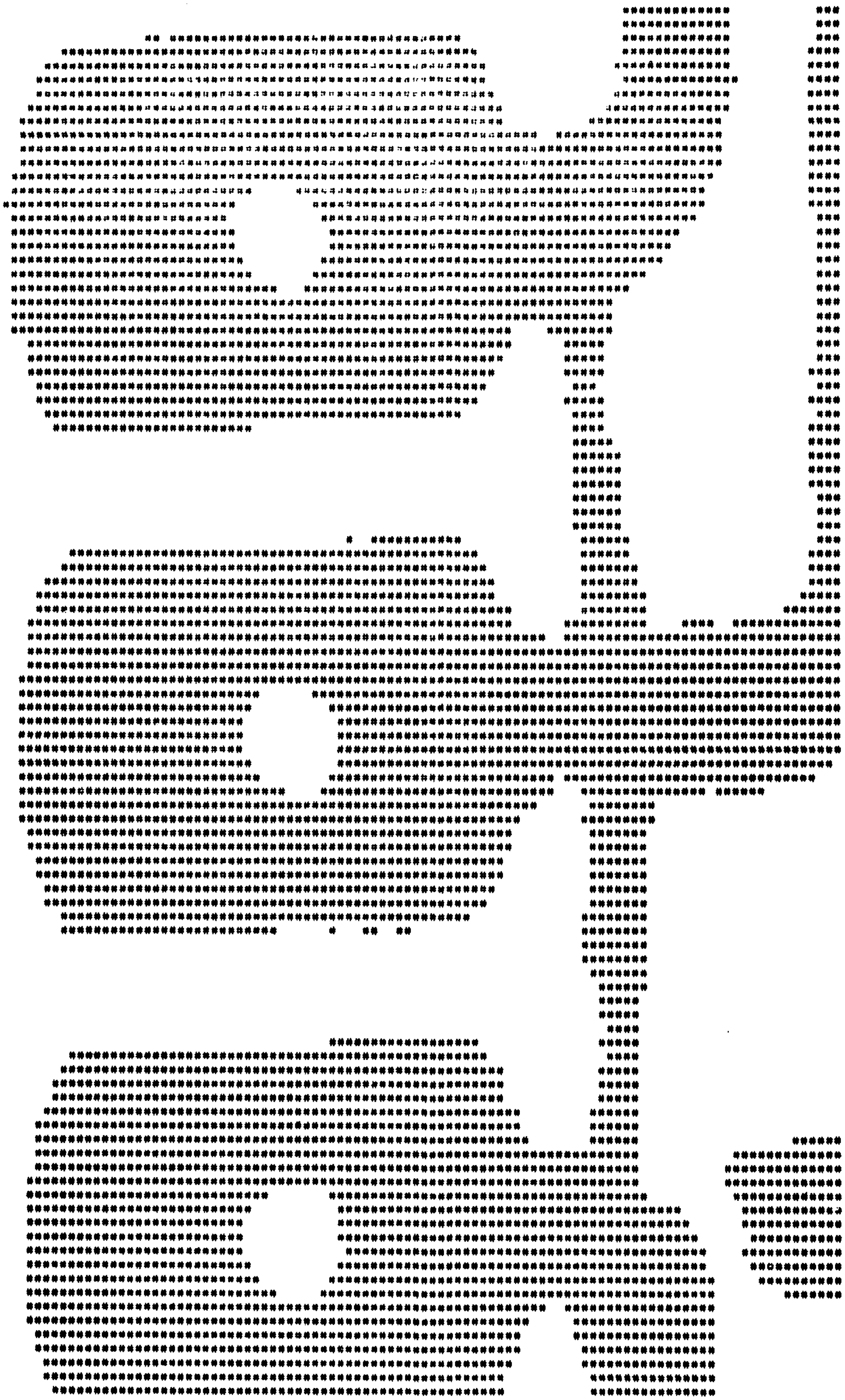


FIGURE 17   SCRATCHED BOARD

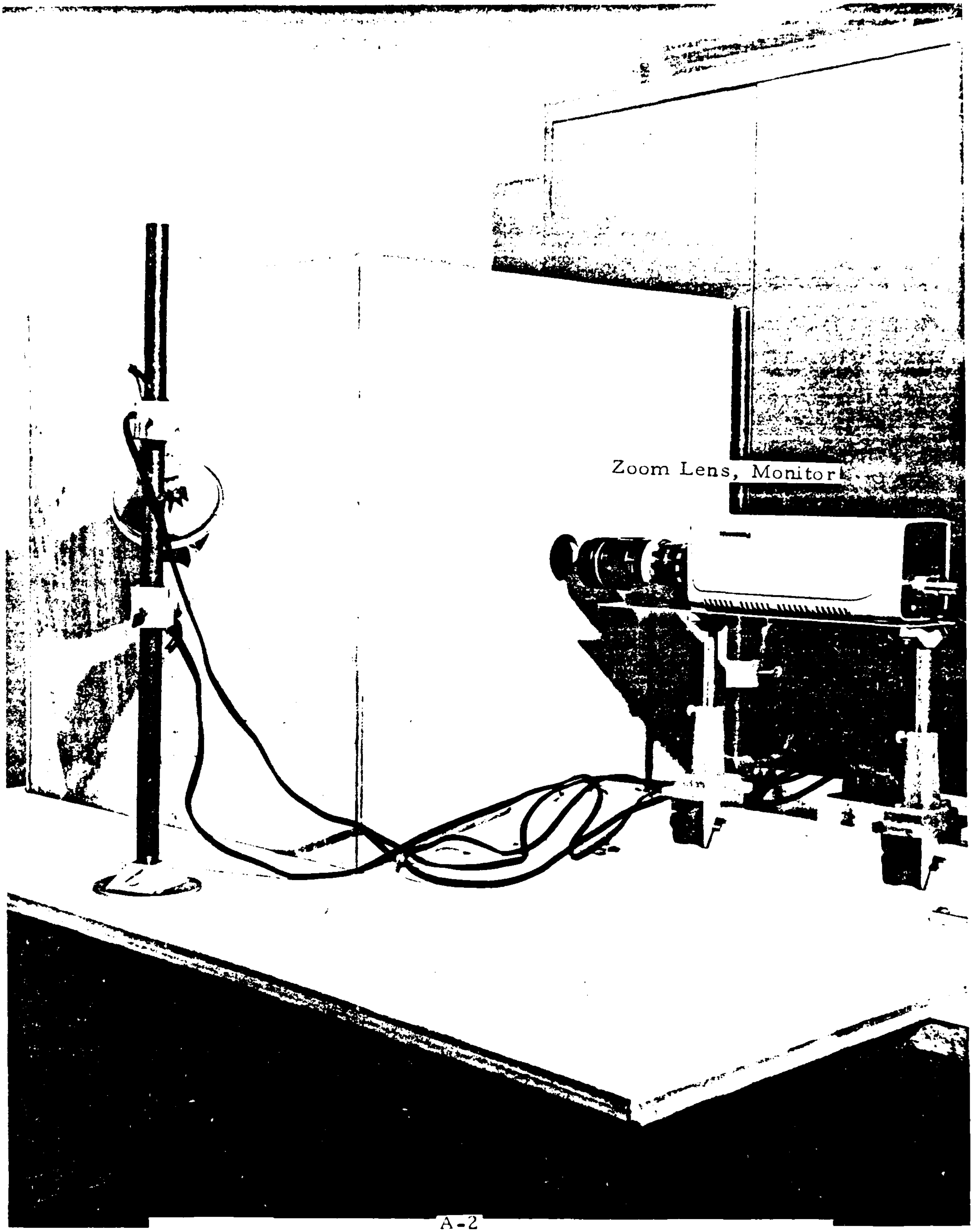


Figure A-2 T.V. Camera, Optical Bench & Light Box

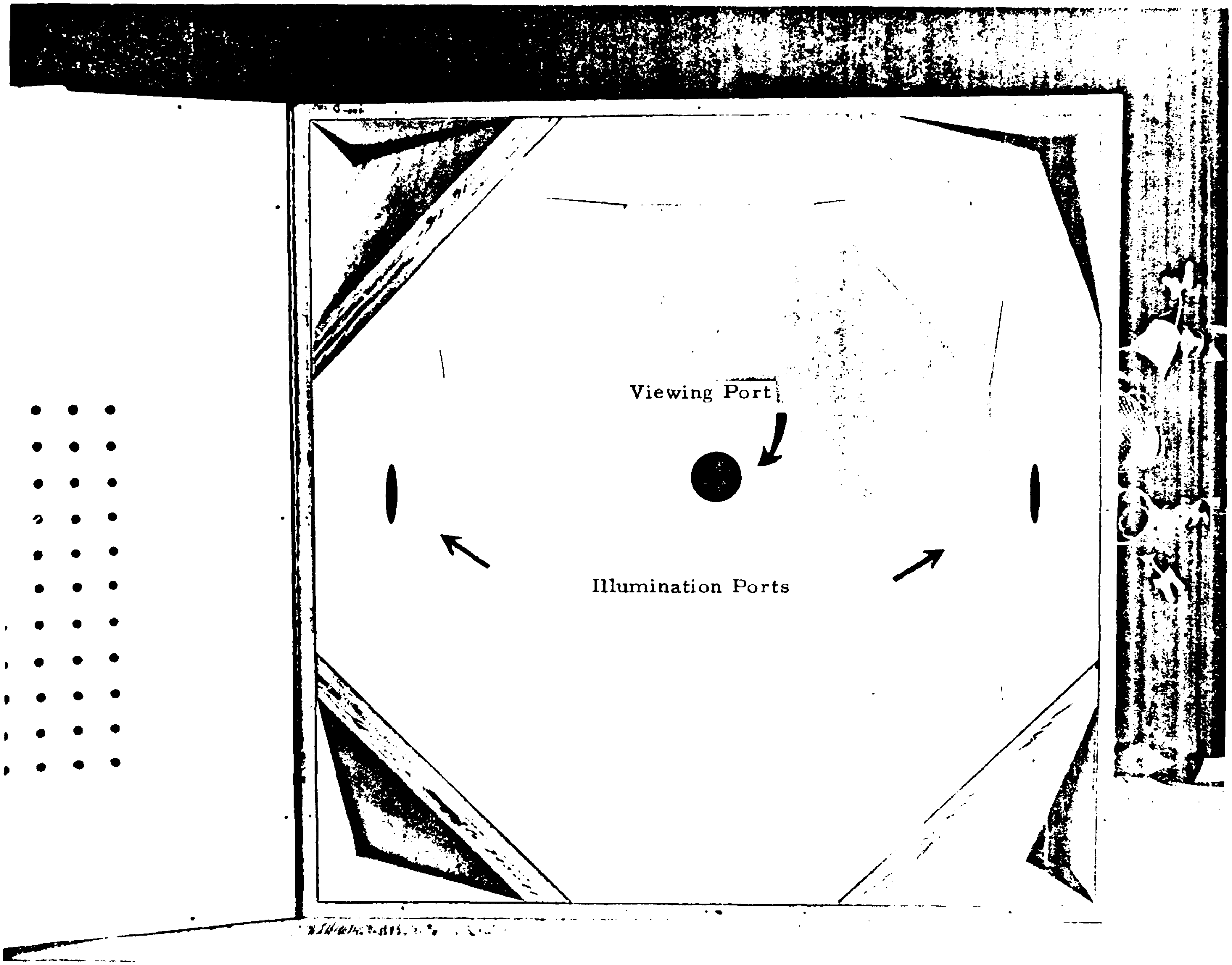


Figure A-3 Internal View of Light Box

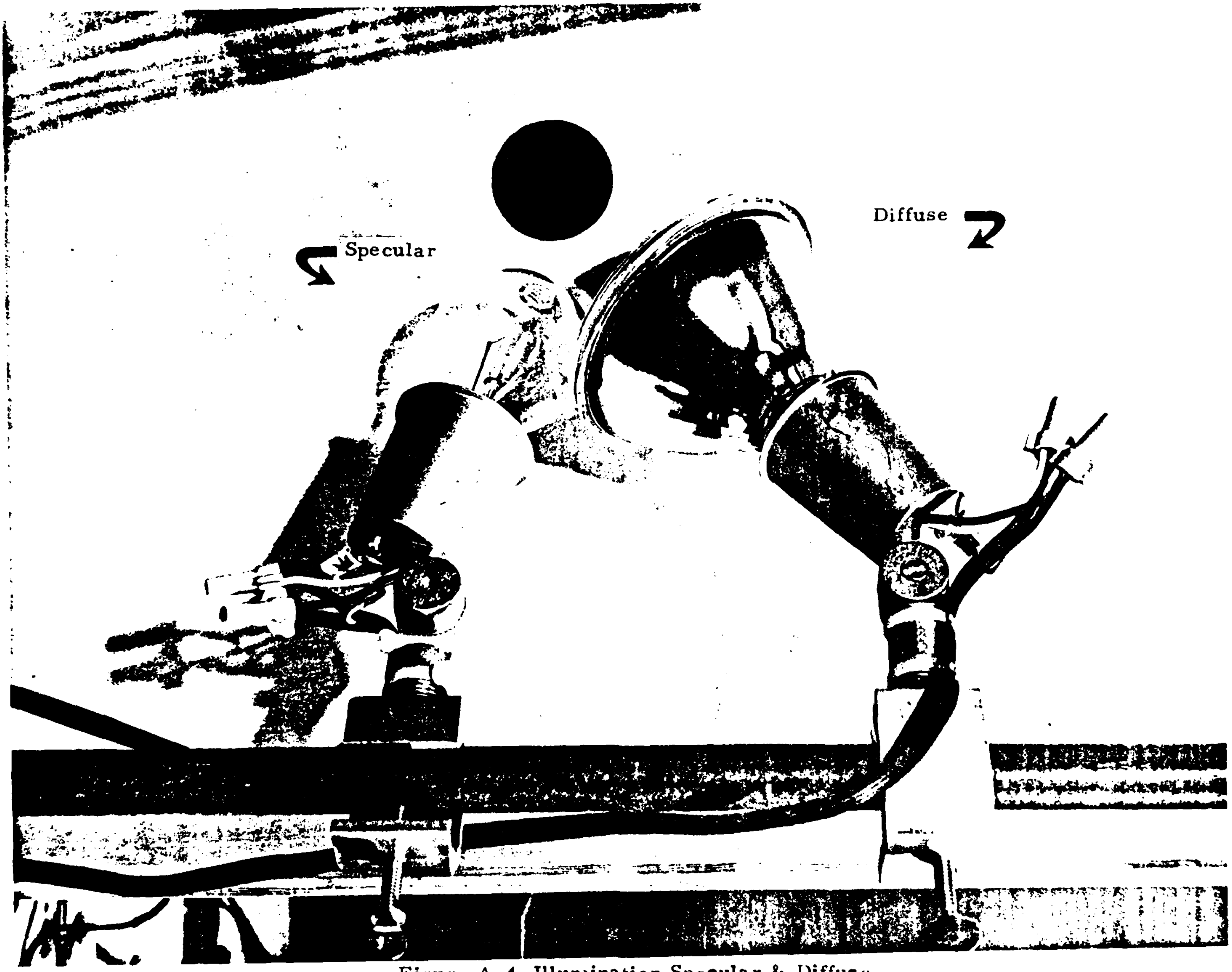


Figure A-4 Illumination Specular & Diffuse

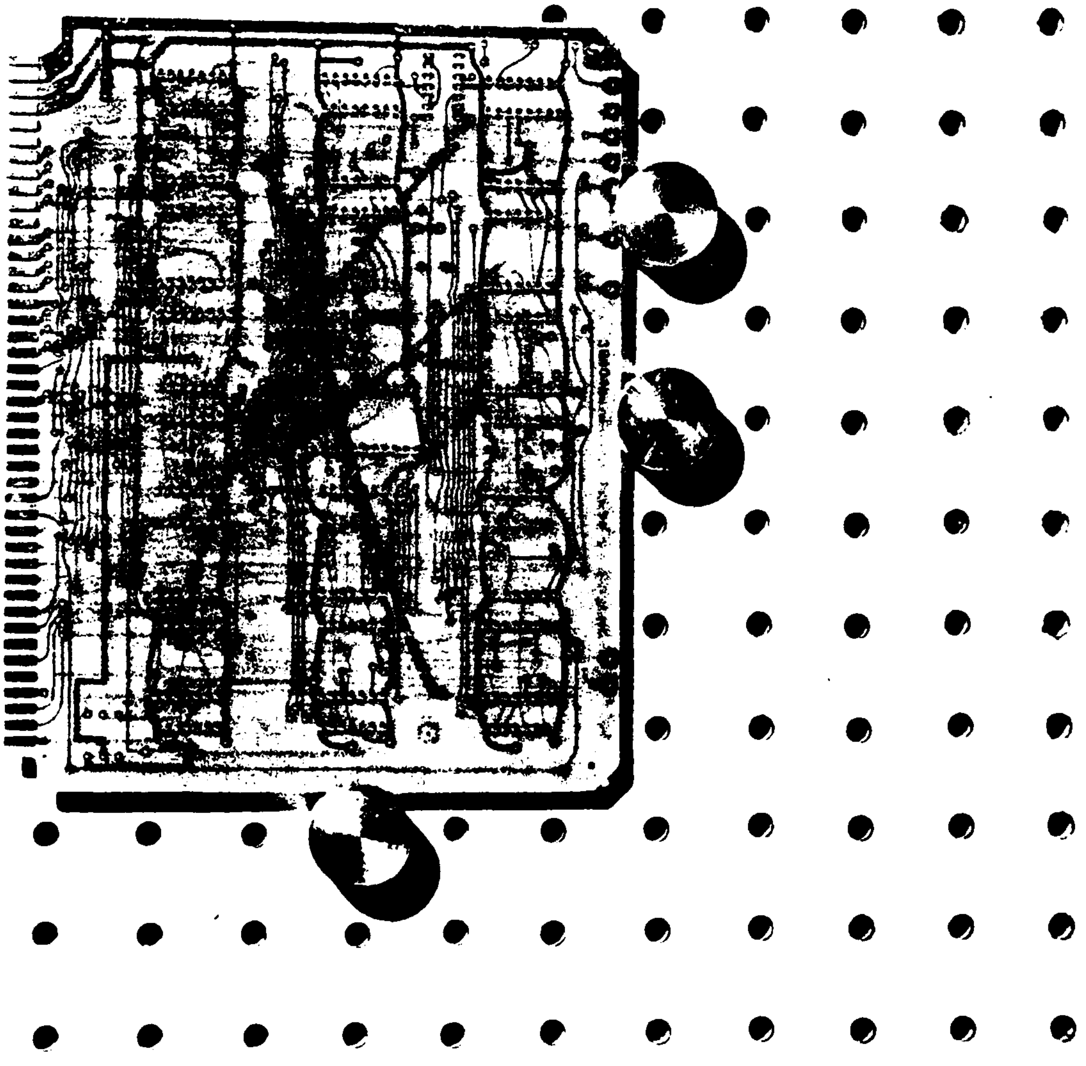


Figure A-5 Test Board on Light Box with Positioning Pegs

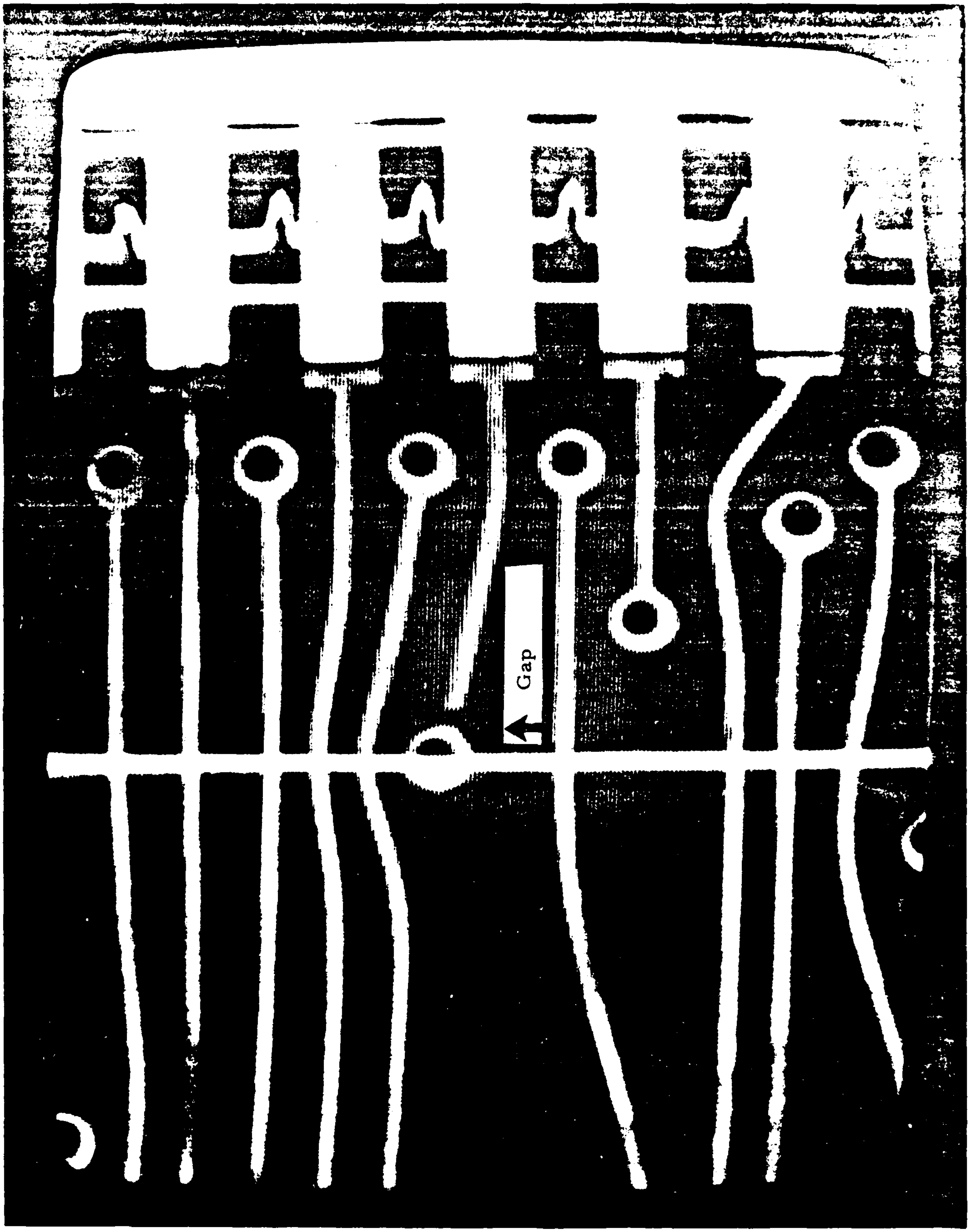


Figure A-6 Monitor Display During Scan of P C Gap

APPENDIX B

DESCRIPTION OF SOFTWARE

THIS IS A DESCRIPTION OF THE SOFTWARE ON THE PRINTED CIRCUIT BOARD INSPECTION PROGRAM. (WO-4502)

THE FOLLOWING PROGRAM IS USED TO GATHER IMAGES FROM THE TELEVISION DIGITIZER SYSTEM.

SCAN2

THIS PROGRAM SCANS A 100 BY 100 BLOCK OF POINTS CENTERED IN THE TELEVISION PICTURE. THE PROGRAM REQUESTS A FILENAME FOR STORAGE ON THE DISK,AND STORES A HISTOGRAM OF THE IMAGE IN THE FIRST BLOCK OF THE FILE. THE DATA IS PACKED TWO BYTES PER WORD IN THE FILE.

THE FOLLOWING PROGRAMS ARE USED TO DISPLAY THE DATA ON THE TEKTRONIX TERMINAL OR THE LINE PRINTER.

HISTOGRAM

THIS PROGRAM DISPLAYS THE HISTOGRAM ON THE SCREEN WITH A SCALE AT THE BOTTOM. WHEN CALLED, IT REQUESTS A FILENAME.

DOTMAT

THIS PROGRAM DISPLAYS THE DATA ON THE TEKTRONIX TERMINAL AS A CARTOON. THE PROGRAM REQUESTS A FILENAME AND THEN REQUESTS THE LOWER AND UPPER LIMITS OF THE VALUES TO BE DISPLAYED.

PRTMAT

THIS PROGRAM IS SIMILAR TO THE ABOVE DOTMAT, EXCEPT IT DISPLAYS THE DATA ON THE LINE PRINTER.

PRTHIST

THIS PROGRAM IS SIMILAR TO THE ABOVE "HISTOGRAM", EXCEPT IT PRINTS THE DATA ON THE LINE PRINTER.

PRTSUM

THIS PROGRAM PRINTS THE SAME DATA AS "PRTHIST", EXCEPT THE DATA IS IN TABULAR FORM. THIS FORMAT USES ONLY ONE PAGE ON THE PRINTER

THE ABOVE PROGRAMS REQUIRE IMAGE FILES THAT WERE SCANNED USING "SCAN2".
[THESE PROGRAMS WERE WRITTEN BY BOB JONES, AND USE THE FOLLOWING SUBROUTINES.]

THE FOLLOWING SUBROUTINES ARE USED IN THE DETAILED DATA HANDLING.

SPOT

THIS PROGRAM IS USED TO INPUT DATA FROM THE CAMERA.

LOGIC

THIS PROGRAM IS USED TO PACK AND UNPACK THE DATA IN ORDER TO CONSERVE STORAGE SPACE.

[THE ABOVE MACHINE LANGUAGE SUBROUTINES (AND MANY EARLIER EXPERIMENTAL PROGRAMS) BY TOMMY REYNOLDS.]

THE FOLLOWING PROGRAMS WERE PREPARED FOR SCI SYSTEMS, INC.
BY BILL POPE, TELCOM DATA CORPORATION.

THERE ARE 3 ENHANCED PCBOARD INSPECTION PROGRAMS, BOARD, CARD, AND PCCARD. ALL PROGRAMS ARE SIMILAR IN FUNCTION. THEY EXAMINE A 100X100 POINT SCAN OF A PCBOARD AND DETERMINE VARIOUS ITEMS CONCERNING THE GEOMETRY OF THE BOARD. THESE INCLUDE OUTLINING ALL RUNS ON THE BOARD, DETERMINING THE SIZE AND LOCATION OF THE RUNS, PROVIDING CALCULATIONS FOR ANY HOLES FOUND. AND EXAMINING THE MINIMUM DISTANCES BETWEEN RUNS.

BOARD

BOARD WILL AUTOMATICALLY TAKE A SERIES OF LIVE SCANS OF A PC BOARD. ALTERNATELY IT WILL TAKE A SINGLE PRE-SCANNED FILE. FOR EACH SCAN, IMAGE THRESHOLDS ARE DETERMINED BY LOOKING AT THE SCAN DATA AS IF IT WERE A CONTOUR MAP. AS THE IMAGE MOVES BETWEEN LEVELS, THE AVERAGE VALUES BETWEEN LEVELS IS DETERMINED AS THE THRESHOLD SEPERATING THE BOARD FROM THE RUN. THESE THRESHOLDS ARE USED FOR RUN DETERMINATION AND HOLE DEFINITION. FINALLY CRITICAL DISTANCES BETWEEN RUNS ARE EXAMINED BY TRACING A CRITICAL DISTANCE BORDER AROUND EACH RUN.

SUBROUTINES USED BY BOARD ARE:
BOARD, AEDGE, TSPOT, LOGIC, UNPACK, SMOOTH, TRACE, FILLIN, INTERNAL, DISTANCE, TRACK, CKPT, CONVAL, SPDOT, SPLIN, SPMOV, HOLE, SPDOTLIN.

PCCARD

PCCARD IS ALMOST THE SAME AS BOARD WITH THESE EXCEPTIONS. FOR LIVE SCANS, PCCARD PROVIDES ONLY A SINGLE SCAN. THE MINIMUM DISTANCE FROM EACH RUN TO ANY OTHER RUN IS DETERMINED AS OPPOSED TO JUST LOOKING AT CRITICAL DISTANCES.

SUBROUTINES USED ARE.
PCCARD, AEDGE, TRACE, SEGMENT, LSCAN, SPOT, UNPACK, SMOOTH, FILLIN, HOLE, INTERNAL, SVDISTANCE, GUESS, CONHIST, LOGIC, SPDOT, SPLIN, SPMOV, SPDOTLIN

THE FOLLOWING SUBROUTINES ARE USED TO DISPLAY GRAPHICS ON THE TEKTRONIX 4006-1 TERMINAL:

SPLIN (IXPOS, IYPOS, IXEND, IYEND) THIS SUBROUTINE IS USED TO PLOT LINES FROM LOCATION (IXPOS, IYPOS) TO (IXEND, IYEND).

SPMOV (IXPOS, IYPOS) THIS SUBROUTINE IS USED TO MOVE THE POSITION OF THE NEXT CHARACTER TO BE PRINTED TO (IXPOS, IYPOS).

SPDOT (IXPOS, IYPOS) THIS SUBROUTINE IS USED TO PLOT A DOT AT POSITION (IXPOS, IYPOS).

[THESE GRAPHICS PROGRAMS WERE WRITTEN BY PETER P. PRYOR JR.]

CARD

CARD WAS THE INITIAL DEVELOPMENT PROGRAM. IT ASLO IS SIMILAR TO THE OTHER TWO PROGRAMS. IT USES ONLY PRESCANNED FILES CREATED BY PROGRAM SCAN2. THE METHOD OF DETERMINING THRESHOLDS IS INSPECTION OF THE HISTOGRAM DATA. THRESHOLDS MUST BE ENTERED MANUALLY AND PLOTTING OF THE HISTOGRAM DATA IS OPTIONAL.

SUBROUTINES USED ARE.
CARD, TRACE, SEGMENT, UNPACK, PEAKS, SMOOTH, PLHIST, EDGE, FILLIN, INTERNAL, SVDISTANCE, GUESS, CONHIST, LOGIC, SPDOT, SPLIN, SPMOV, HOLE, SPDOTLIN

TWO OTHER PROGRAMS, BOTH INCONSEQUENTIAL, WERE LEFT ON THE FILE. CONTOUR AND BVIEW PROVIDE A 3D CONTOUR PLOT AND BACK-VIEW OF PRESCANNED DATA FILES.

```
C       PROGRAM PCCARD
C
C       PREPARED FOR SCI SYSTEMS, INC.
C        BY BILL POPE, TELCOM DATA CORPORATION.
C
C       THIS SUBROUTINE IS THE MAIN DRIVER PROGRAM WHICH EXAMINES SCANS
C       OF PC BOARDS.
C
C
C
C
        COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
        INTEGER ARRAY(100,100), SUM(256), AMAX(5,20)
        EXTERNAL OVER0,OVER1,OVER3,OVER6,OVER7
C
C
C       INITIALIZE OVERLAYS
C
        CALL OVOPN(5,"PCCARD.OL",IERR)
        IF(IERR.EQ.1)GO TO 880
        TYPE " OVERLAY OPENING ERROR ",IERR
        STOP
880     CONTINUE
C
C
        LP=10
        TYPE " PRINT OR TYPE RESULTS? P,T "
        READ(11,100)NANS
        IF(NANS.NE."P")GO TO 7
        LP=12
        CALL FOPEN(12, "$LPT" )
7       CONTINUE
C
C
C
C       FIND OUT ABOUT LIVE SCAN
C
10      TYPE " LIVE SCAN? Y,N"
        READ(11,100)NANS
100     FORMAT(S1)
        IF(NANS.EQ."N")GO TO 20
        IF(NANS.NE."Y")GO TO 10
C
C       LOAD SCAN OVERLAY
C
        CALL OVLOD(5,OVER7,0,IERR)
        IF(IERR.NE.1)GO TO 980
C
        CALL LSCAN
C
```

```
      GO TO 30
C
C
 20   CONTINUE
C
C
C     INITIALIZE AND UNPACK DATA
C
      CALL OVLOD(5,OVER0,0,IERR)
      IF(IERR.NE.1)GO TO 980
C
      CALL UNPACK
C
30    CONTINUE
      CALL OVLOD(5.OVER1,0,IERR)
      IF(IERR.NE.1)GO TO 980
C
C
C     SMOOTH MAKES EACH POINT EQUAL THE AVERAGE OF ITS NEIGHBORS
C
      CALL SMOOTH
C
C
C
C
C
C     PEAK TAKES HISTOGRAM IN SUM AND DETERMINES PEAKS.  FIRST MAJOR
C     PEAK FOUND LIMITS ARE RETURNED IN LTH AND UTH.
C
C     CALL PEAKS
C
C
C     TEST THRESHOLD
C
      CALL OVLOD(5,OVER6,0.IERR)
      IF(IERR.NE.1)GO TO 980
C
C
      CALL CONHIST(ARRAY,LP,LTL,LTU)
C
C
C
C
C
C     CALL EDGE TO BREAK ARRAY INTO AREAS WITHIN LTH, UTH
C
      CALL AEDGE(LTL,LTU)
C
C
C
```

```
C
         CALL OVLOD(5.OVER3,0,IERR)
         IF(IERR.NE.1)GO TO 980
C
C
C        NOW TRY TO FIND DISTANCES BETWEEN AREAS
C
C
         CALL DISTANCE
C
C
         CALL CLOSE(5,IERR)
         CALL RESET
         STOP
980      TYPE " ERROR LOADING OVERLAY ",IERR
         STOP
         END
```

```
          TITL  LOGIC                  ;
,        DGC FORTRAN LOGIC                          ;
.        FUNCTIONS                     ;
                                       ;
          ENT     IAND, IOR, ILEFT     ,
         .ENT     IRIGHT               ;
                                       ;
         .EXTN    FRET                 ;
         .EXTD    .CPYLS               ;
                                       ;
VAL      =        -167                 ; FUNCTION VALUE RETURNED
ARG1     =        VAL+1                ; PARAMETER 1
ARG2     =        ARG1+1               ; PARAMETER 2
 SIZE    =        2                    ; FRAME STACK SIZE
         .NREL                         ; NORMAL MEMORY
                                       ;
.        IAND(I, J)                    ;
         .SIZE                         ;
IAND:    JSR      @.CPYLS              ; COPY ARGUMENT ADDRESSES
         LDA      0 @ARG1 3            ; GET PARAMETER 1
         LDA      1 @ARG2 3            ; GET PARAMETER 2
         AND      1 0                  ; PERFORM LOGICAL AND
         STA      0 @VAL 3             ; SET FUNCTION VALUE
         FRET                          ; RETURN TO CALLER
                                       ;
;        IOR(I, J)                     ;
                                       ;
         .SIZE                         ;
IOR:     JSR      @ CPYLS              ; COPY ARGUMENT ADDRESSES
         LDA      0 @ARG1 3            ; GET VARIABLE
         LDA      1 @ARG2 3            ; GET PARAMETER 2
         COM      1 1                  ; FORM 1@S COML
         AND      1 0                  ; MASK OUT BITS
         ADC      1 0                  ; INSERT BITS
         STA      0 @VAL 3             ; SET RETURN VALUE
         FRET                          ; RETURN TO CALLER
                                       ;
;        ILEFT(I, NUMBER) ;
;        IRIGHT(I, NUMBER)             ;
                                       ;
         .SIZE                         ;
ILEFT    SUB      2 2                  ; INDICATE LEFT
         JMP      COMMN                , GO TO COMMON ROUTINE
          SIZE                         ;
IRIGHT.ADC        2 2                  . INDICATE RIGHT
COMMN:   JSR      @.CPYLS              , COPY ARGUMENT ADDRESSES
         LDA      0 @ARG1 3            ; GET PARAMETER 1
         LDA      1 @ARG2 3            ; GET SHIFT COUNT
         NEG      1 1 SNR              ; NEGATE COUNT
         JMP      OUT                  , RETURN IF ZERO
```

```
LOOP    MOV      2 2 SNR        ; SKIP IF RIGHT
        MOVZL    0 0 SKP        ; SHIFT LEFT
        MOVZR    0 0            ; SHIFT RIGHT
        INC      1 1 SZR        ; SKIP IF DONE
        JMP      LOOP           ; SKIP SOME MORE
OUT:    STA      0 @VAL 3       ; SET OUTPUT VALUE
        FRET                    ;
        .END
```

```
      LTHF=LTH+NBIAS
      UTHF=UTH+NBIAS
     *PAREA=0.0
      JMIN1=JMIN+1
      IF(JMIN1.GT.JMAX)RETURN
      DO 150 J=JMIN1,JMAX
C
C     INITIALIZE FLAG.  AS WE SEARCH THIS PORTION OF ARRAY, POINTS MAY
C     FALL INSIDE THRESHOLD WHILE BEING EXTERNAL TO SEGMENT WE ARE
C     EXAMINING.  WILL SET FLAG WHEN WE FIND FIRST MARKED EDGE TO
C     INDICATE MOVING INTERNAL TO SEGMENT.
C
      NFL=0
      DO 149 I=IMIN,IMAX
      IF(NFL.GT.0) GO TO 145
C
C     IS THIS LEADING EDGE OF SEGMENT.
C
      IF(A(I,J).GE.NBIAS)NFL=1
      GO TO 147
C
C     DETERMINE IF POINT IS WITHIN THRESHOLDS.
C
145   IF(A(I,J).LT.LTH.OR.A(I,J).GT.UTH)GO TO 146
C
C
C     FOUND A POINT.  LOOK AT ADJACENT POINTS TO SEE IF ANY
C     ARE OUT OF THRESHOLD LIMITS.  IF SO, CONSIDER AS
C     START OF HOLE IN AREA.  GO TO TRACE TO FIND EDGE OF HOLE.
C     LOOKING AHEAD TO NEXT ROW AND COLUMN TO "ANTICIPATE" HOLES.
C
      N=J
      M=I-1
      IF(M.LT.1)GO TO 12
      IF(A(M,N).GE.LTH.AND.A(M,N).LE.UTH)GO TO 12
      IF(A(M,N).GE.LTHF)GO TO 12
      GO TO 25
C
12    M=I+1
      IF(M.GT.100) GO TO 14
      IF(A(M,N).GE.LTH.AND.A(M,N).LE.UTH)GO TO 14
      IF(A(M,N).GE.LTHF)GO TO 14
      GO TO 25
C
14    M=I
      N=J-1
      IF(N.LT.1) GO TO 16
      IF(A(M,N).GE.LTH.AND.A(M,N).LE.UTH)GO TO 16
      IF(A(M,N).GE.LTHF)GO TO 16
      GO TO 25
```

```
        OVERLAY OVERA
        SUBROUTINE FILLIN(LTH,UTH,NBIAS,RAREA,IDEBUG,B,NBP,NHOLE,NSPOT)
C
C       PREPARED FOR SCI SYSTEMS, INC
C        BY BILL POPE, TELCOM DATA CORPORATION.
C
C       THIS SUBROUTINE WORKS WITH AN AREA WHOSE EDGES HAVE BEEN DEFINED IN
C       SUBROUTINE EDGE. GIVEN THAT THE AREA FALLS BETWEEN THE GIVEN
C       I AND J MINS AND MAX VALUES, AND THAT THE EDGE POINTS HAVE BEEN
C       GIVEN A BIAS OF NBIAS, THIS SUBROUTINE FILLS IN ALL THE POINTS INTERNAL
C       TO THE AREA (FALLING, OF COURSE, BETWEEN THE SEARCH LIMITS).
C       IT ALSO CALCULATES THE AREA FOUND BY SUMMING THE NUMBER
C       OF POINTS.
C       WHILE PERFORMING THE INTERNAL SEARCH FOR POINTS WITHIN THE SEGMENT,
C       IF A POINT IS FOUND WHICH IS OUTSIDE THE SEARCH LIMITS THEN IT IS
C       ASSUMED TO BE THE EDGE OF AN INTENAL "HOLE". THE SUBROUTINE HOLE
C       IS CALLED TO TRACE THE HOLE BOUNDARIES AND DETERMINE HOLE RELATED
C       PARAMETERS.
C
C
C       VARIABLES USED INCLUDE:
C
C       LTH,UTH         LOWER AND UPPER THRESHOLDS FOR INTERNAL SEARCH
C       NBIAS           THE VALUE USED TO "MARK" THE EDGES OF THIS SEGMENT.
C                       (MARK BY ADDING NBIAS TO ARRAY(I,J))
C       RAREA           THE CALCULATED INTERNAL AREA OF THE SEGMENT. VALUE
C                       IS REAL NUMBER AND DOES NOT INCLUDE HOLE AREAS.
C       B(2,NPT)        INTEGER ARRAY OF 2,NPT POINT CONTAINING THE I,J
C                       VALUES OF THIS SEGMENTS EDGE.
C       NHOLE           NUMBER OF HOLES FOUND.
C       NSPOT           NUMBER OF SPOTS FOUND (HOLE LESS THAN 20 PERIMETER).
C       IMIN,IMAX       THE MINIMUM AND MAXIMUM COORDINATES OF
C       JMIN,JMAX               THE CURRENT SEGMENT.
C
C
C
        COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
        COMMON /PLT/ ISX,ISY,MIDX,MIDY,NXS,NYS
        INTEGER ARRAY(100,100),SUM(256),AMAX(5,20)
        INTEGER A(100,100),UTH,UTHF
        INTEGER B(2,1500)
        EQUIVALENCE (A(1,1),ARRAY(1,1))
C
C
        NHOLE=0
        NSPOT=0
        IA=0
C
C       SET PSUEDO THRESHOLDS FOR POINTS "MARKED" BY THIS NBIAS
C
```

```
C
C       SAVE MINIMUM AND MAXIMUM VALUES FOR SUBROUTINE DISTANCE.
C
        AMAX(1,NCNT)=IMIN
        AMAX(2,NCNT)=IMAX
        AMAX(3,NCNT)=JMIN
        AMAX(4,NCNT)=JMAX
        AMAX(5,NCNT)=NHO
78      CONTINUE
79      CONTINUE
        MCNT=NCNT
        RETURN
 880    TYPE " SECOND LEVEL OVERLAY LOAD ERROR"
        STOP
        END
```

```
C
C
C
C      NOW CALL SUBROUTINE TRACE TO TRACE OUT THE EDGE.
C
       NSPL=256
C      MARK AS OUTTER EDGE OF AREA
C
       CALL TRACE(IROW,JCOL,IO,JO,LTH,UTH,PERIM,AREA,NSPL,
     1 IMIN,IMAX,JMIN,JMAX,B,NPT)
       IF(PERIM.NE.0.0)GO TO 77
C
C      ONLY A SINGLE POINT SO BACK OFF AREA.
       AREA=AREA-NAREA
       GO TO 78
77     CONTINUE
C
C
C      NOW WE HAVE THE BOUNDARY, FILL IN THE REST
C
C
C
C
       NCNT=NCNT+1
       CALL FILLIN(LTH,UTH,AREA,RAREA,IDEBUG,B,NPT,NHO,NSPT)
C
C
C      GO LOOK AT AREAS INTERNAL DIMENSIONS
C
C
C
       CALL OVLOD(5,OVERB,0,IER)
       IF(IER.NE.1)GO TO 880
       CALL INTERNAL(B,NPT,XDIM)
C
C
       WRITE(LP,260)NCNT,LTH,UTH
260    FORMAT("0AREA : ",I3,"    BRIGHTNESS BETWEEN ",I4," AND ",I4)
       WRITE(LP,261)PERIM,RAREA
261    FORMAT("      PERIMETER= ",F8.2,5X,"INTERNAL AREA= ",F8.2)
       WRITE(LP,162)XDIM
162    FORMAT("      MINIMUM INTERNAL THICKNESS= ",F6.2)
       WRITE(LP,262)IMIN,IMAX,JMIN,JMAX
262    FORMAT("      IMIN=",I4,"   IMAX=",I4,"   JMIN=",I4,"   JMAX=",I4)
       WRITE(LP,263)NHO,NSPT
263    FORMAT("      AREA CONTAINED ",I3," HOLES AND ",I3," SPOTS.")
C
C
C
C
```

```
C       PEAKS FOUND FROM EXAMINATION OF THE "HISTOGRAM".  WHEN
C       SEARCHING FOR THE EDGES OF AREAS, A FUDGE FACTOR  WILL BE
C       USED TO EXTEND THE SEARCH RANGE.  THE POINTS WITHIN THE
C       EXTENDED RANGE WILL ONLY BE ACCEPTED IF THEY ARE ADJACENT
C       TO A POINT WITHIN THE ORIGINAL SEARCH RANGE.
C
C
        ND=(UTH1-LTH1)/10
        LTH=LTH1-ND
        LTH=0
        IF(LTH.LT.0)LTH=0
        UTH=UTH1+ND
        IF(UTH.GT.255) UTH=255
C
C
C       SIZE FACTORS TO ALLOW FOR THE FACT SCAN IS NOT SQUARE.
C
        NXSIZE=5
        NYSIZE=4
        MIDX=512
        MIDY=370
        ISTARTX=MIDX-((NXSIZE*100)/2)-NXSIZE
        ISTARTY=MIDY+((NYSIZE*100)/2)+NYSIZE
        ISTARTX=200
        ISTARTY=620
C
C       LOOP THROUGH ENTIRE ARRAY
C
        DO 79 JC=1,100
        DO 78 IR = 1,100
        JCOL=JC
        IROW=IR
C       IS DATA POINT WITHIN ORIGINAL THRESHOLDS
        IF(A(IROW,JCOL).LT.LTH1.OR.A(IROW,JCOL).GT.UTH1) GO TO 78
C
C       YES, FOUND FIRST POINT OF NEW AREA
C       BACK UP ON I FOR EXTENDED SEARCH RANGE
C
9       IF(IROW.EQ.1)GO TO 10
        I=IROW-1
        IF(A(I,JCOL).LT.LTH.OR.A(I,JCOL).GT.UTH)GO TO 10
        IROW=I
        GO TO 9
10      CONTINUE
        IO=IROW-1
        JO=JCOL
        AREA=AREA+NAREA
C
        CALL OVLOD(5,OVERA,0,IER)
        IF(IER.NE.1)GO TO 880
```

```
      SUBROUTINE AEDGE(LTH1,UTH1)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C     THIS SUBROUTINE EXAMINES THE SCAN DATA FOUND IN ARRAY A.
C     ALL POINTS WHOSE VALUE IS BETWEEN THE LOWER THRESHOLD LTH
C     AND THE UPPER THRESHOLD UTH ARE DIVIDED INTO SEGMENTS.
C     A SEGMENT IS AN AREA OF POINTS CONTIGUOUS TO EACH OTHER BUT
C     SEPERATED FROM ALL OTHER POINTS WITHIN THE THRESHOLD LIMITS.
C
C
C     A IS AN ARRAY DIMENSIONED 100 BY 100 CONTAINING SCAN DATA POINTS.
C     LTH1 IS THE LOWER BRIGHTNESS THRESHOLD
C     UTH1 IS THE UPPER BRIGHTNESS THRESHOLD
C     NCNT IS THE NUMBER OF AREAS FOUND.
C
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
      COMMON /PLT/ISTARTX,ISTARTY,MIDX,MIDY,NXSIZE,NYSIZE
      INTEGER ARRAY(100,100), SUM(256),AMAX(5,20)
      INTEGER B(2,1500),SEG(3,20)
      INTEGER UTHF,UTH1
      INTEGER A(100,100), AREA, UTH
      EQUIVALENCE (A(1,1),ARRAY(1,1))
      EXTERNAL OVERA,OVERB
C
C
C     OTHER VARIABLES USED ARE:
C       MI - MAXIMUM I (ROW) WITHIN CURRENT COLUMN WHICH IS PART OF AREA
C       LI - LOWEST I ( ROW) WITHIN CURRENT COLUMN
C       NLI - LOWEST I IN LAST COLUMN
C       IS - START I VALUE FOR EXAMINING CURRENT COLUMN
C
C      INITIALIZE DATA. NAREA IS SET TO VALUES THAT WILL
C     UTILIZE THE HIGH ORDER BITS OF A SCAN POINT AS A FLAG
C     INDICATING THE AREA NUMBER TO WHICH IT BELONGS.
C
      AREA=0
      NCNT=0
      NAREA=1024
      IDEBUG=1
      ISP=288
      IUS=287
C
C
      IF((LTH1+UTH1).EQ.0)RETURN
C
C
C     LTH1 AND UTH1 REPRESENT THE NOMINAL DIVISION VALUES BETWEEN
```

```
25      CONTINUE
C
C
C       THATS ALL THE POINTS.  FIND AVERAGE FOR UPPER LIMITS.
C
        LT=IFIX(AVG/CNT)
        WRITE(LP,100)MIN,LT
100     FORMAT("OCONTOUR THRESHOLDS BETWEEN ", I4," AND ",I4)
      ACCEPT"LOWER THRESHOLD ? ",MIN
      ACCEPT"UPPER THRESHOLD ? ",LT
        RETURN
        END
      NBOT=-1
      IB=0
      IT=0
      DO 24 I=1,LPEND
C
C     FIRST CHECK NEW POINT FOR MAXIMUM BRIGHTNESS
C
      IF(ARRAY(I,J).LT.MIN)MIN=ARRAY(I,J)
      NS=ARRAY(I,J)-ARRAY(I+STP,J)
C
C     JUMP DEPENDENT ON WHICH REGION WE HAVE BEEN IN.
C
      IF(SLOPE)30,40,50
C
C     HAVE BEEN SEARCHING ALONG LEVEL (SLOPE = 0).
C     IF THE ABSOLUTE VALUE OF THE SLOPE EXCEEDS THE LIMIT NSTP,
C     THEN CONSIDER STARTING MOVE TO NEXT LEVEL.
C
C
C     SAVE THE PRESENT VALUE IN NTOP OR NBOT DEPENDENT ON WHICH
C     DIRECTION WE ARE STARTING TO MOVE.
C
40    IF(IABS(NS).LT.NSTP)GO TO 24
      IF(NS.GT.0)GO TO 45
      NTOP=ARRAY(I,J)
      GO TO 20
45    NBOT=ARRAY(I,J)
      GO TO 20
C
C
C     AT LABELS 30 AND 50, WE HAVE BEEN MOVING BETWEEN LEVELS.  WHEN THE
C     VALUE OF THE SLOPE FALLS BELOW THE LIMIT NSTP, THINGS ARE "LEVELING"
C     OUT.  IF THERE WAS A PREVIOUS LEVEL (NOT STARTING A SIDE OF SCAN),
C     THEN GET AVERAGE BRIGHTNESS AND SAVE.
C
30    IF(NS.LT.-NSTP)GO TO 24
      IF(NTOP.LT.0)GO TO 31
      AVG=AVG+(FLOAT(NTOP+ARRAY(I,J)))/2.
      CNT=CNT+1
31    NTOP=-1
      NBOT=-1
      GO TO 20
50    IF(NS.GT.NSTP)GO TO 24
      IF(NBOT.LT.0)GO TO 31
      AVG=AVG+(FLOAT(ARRAY(I,J)+NBOT))/2.
      CNT=CNT+1
      GO TO 31
 20   SLOPE=NS
      IF(IABS(NS).LT.NSTP)SLOPE=0
24    CONTINUE
```

```
      OVERLAY OVER6
      SUBROUTINE CONHIST(ARRAY,LP,MIN,LT)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C
C     THIS SUBROUTINE TRIES A DIFFERENT METHOD FOR THRESHOLDS.
C     ASSUMMING THAT THE DATA ARRAY RECEIVED FROM A SCAN BASICALLY
C     FORMS A CONTOUR OF THE BRIGHTNESSES, CONHIST SEARCHES ONE ROW
C     AT A TIME LOOKING FOR THE CHANGES IN CONTOUR. THE CHANGING CONTOUR
C     CAN THEN BEE SEEN AS LEVELS REPRESENTING THE BOARD, THE RUNS, AND
C     THE "BOTTOM" OF HOLES. BY FINDING THE AVERAGE MIDPOINT BETWEEN LEVELS
C     EACH TIME THE CONTOUR MOVES, THESE MIDPOINTS CAN THEN BE
C     AVERAGED TO FIND THE "UPPER", OR DARKER, THRESHOLD. AT THE
C     SAME TIME, THE BRIGHTEST SPOT ON THE BOARD CAN BE FOUND TO USE
C     AS THE LOWER THRESHOLD.
C
C
C
      INTEGER ARRAY(100,100),STP,SLOPE
C
C
C     THE METHOD FOR DETECTING MOVEMENT BETWEEN LEVELS IS EXAMINATION OF THE
C     SLOPE OF THE CONTOUR. SINCE THE "X" COORDINATE WILL BE CONSTANT FOR
C     EACH COMPARISON, ONLY THE "Y" COORDINATE (DIFFERENCE IN BRIGHTNESS)
C     WILL BE EXAMINED.
C
C
C     INITIALIZE VALUES
C
      NSTP=10
      STP=1
      LPEND=100-STP
      MIN=255
      CNT=0.
      AVG=0.
      DO 25 J=5,100,5
C
C     AS WE EXAMINE EACH ROW, NS REPRESENTS THE NEW SLOPE FOR THE NEXT
C     POINT TO EXAMINE WHILE SLOPE IS THE GENERAL SLOPE IN THE REGION
C     WE ARE SEARCHING. A SLOPE OF ZERO INDICATES WE ARE ON THE BOARD,
C     A NEGATIVE SLOPE INDICATES MOVEMENT TOWARDS A BRIGHTER LEVEL, AND
C     A POSITIVE SLOPE TOWARD A DARKER LEVEL.
C
C
      NS=ARRAY(1,J)-ARRAY(STP+1,J)
      SLOPE=NS
      IF(IABS(NS).LT.NSTP)SLOPE=0
      NTOP=-1
```

```
            IF(N.GT.0.AND.N.LT.257)GO TO 7
            TYPE " ERROR IN DATA- ",I,J,N
            N=256
7           CONTINUE
            SUM(N)=SUM(N)+1
10          CONTINUE
11          CONTINUE
C
C
            RETURN
            END
```

```
      OVERLAY OVER1
      SUBROUTINE SMOOTH
C
C     THIS SUBROUTINE TAKES THE 100 BY 100 SCAN DATA IN ARRAY
C     AND SMOOTHS IT BY REVALUING EACH POINT TO THE AVERAGE
C     OF IT'S EIGHT ADJACENT NEIGHBORS.  A NEW 256 VALUE
C     "HISTOGRAM" IS THEM CONSTRUCTED IN SUM.
C
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
      INTEGER AMAX(5,20)
      INTEGER ARRAY(100,100), SUM(256)
C
      TYPE " SMOOTH DATA? Y,N"
      READ(11,100)NANS
 100  FORMAT(S1)
      IF(NANS.EQ."N")RETURN
C
C     CLEAR OLD "HISTOGRAM" AS NEW ONE WILL BE FORMED.
C
      DO 9 I=1,256
9     SUM(I)=0
C
C
C     AVERAGE ARRAY POINTS.
C
      DO 11 J=1,100
      DO 10 I=2,99
      ISUM=0
      ISUM=ARRAY(I+1,J)+ARRAY(I-1,J)
      NN=J-1
      NS=I--1
      NE=I+1
      IF(J.EQ.1)GO TO 27
      DO 17 K=NS,NE
      ISUM=ISUM+ARRAY(K,NN)
17    CONTINUE
27    CONTINUE
      NN=J+1
      IF(J.EQ.100)GO TO 28
      DO 18 K=NS,NE
      ISUM=ISUM+ARRAY(K,NN)
18    CONTINUE
28    ND=8
      IF(J.EQ.1.OR.J.EQ.100)ND=5
      ARRAY(I,J)=ISUM/ND
C
C     NOW USE NEW VALUE FOR "HISTOGRAM".
C
      N=ARRAY(I,J)+1
```

```
        LOW = LOW + 1
        M = 0
        IF (DELTAY.LT.STEP2) GO TO 11
        DELTAY = 0
        LOW = LOWERY
        M = 0
10      CONTINUE
C
C       END OF SCAN.  MAKE SURE ALL POINTS PUT IN ARRAY.
C
C
        IF(IA.GT.100)RETURN
        IF(K.EQ.0)RETURN
C
C
        DO 50 I=1,K
        ARRAY(IA,JA)=POINTS(3,I)
        JA=JA+2
        IF(JA.LE.100)GO TO 50
        JA=JA-99
        IF(JA.EQ.2)GO TO 50
        JA=1
        IA=IA+1
 50     CONTINUE
C
C
        RETURN
        END
```

```
C
C         LOOP TO CREATE 500 POINTS TO SPOT
C
          IA=1
          JA=1
C
C
          DO 10 COLUMN = LEFTX, RIGHTX, STEP
11        DO 20 ROW = LOW, UPPERY, STEP2
          K = K + 1
          POINTS(1, K) = COLUMN
          POINTS(2, K) = ROW
          M = M + 1
          IF (K. EQ. 500) GO TO 30
          GO TO 20
C
C         CALL SPOT
C
30        CALL SPOT(K, POINTS)
C
C
C
C
C
C
C         PUT DATA INTO CORE ARRAY
C
C
          DO 45 I=1,500
          ARRAY(IA,JA)=POINTS(3,I)
          JA=JA+2
          IF(JA. LE. 100)GO TO 45
          JA=JA-99
          IF(JA. EQ. 2)GO TO 45
          JA=1
          IA=IA+1
 45       CONTINUE
C
C
C
C
C
          K = 0
          GO TO 21
20        CONTINUE
C
C         NOTE THAT THERE IS A DOUBLE DO LOOP USING STATEMENT NUMBERS 10 AND 20
C         CARE SHOULD BE TAKEN IN FOLLOWING THE CORRECT PATH
C
21        DELTAY = DELTAY + STEP
```

```
      OVERLAY OVER7
      SUBROUTINE LSCAN
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C
C     THIS PROGRAM IS NAMED LSCAN
C     BASICALLY AN ARRAY CALLED POINTS IS CREATED THRU A DO LOOP
C     AND PASSED TO AN ASSEMBLY LANGUAGE SUBROUTINE CALLED SPOT
C     WHICH DOES THE ACTUAL SCANNING.  THE THIRD ARGUMENT OF THE
C     THREE DIMENSIONAL ARRAY CONTAINS THE RETURNED DIGITIZED
C     VALUE WHICH CAN VARY FROM 0 TO 255.  THE 500 VALUES
C     RETURNED ARE PLACED IN THE DATA ARRAY(100,100).  WHEN
C     IT IS FULL, CONTROL IS RETURNED TO THE DRIVER.
C
C
C     LEFTX IS THE X OF THE BEGINNING (X,Y) POSITION
C     RIGHTX IS THE X OF THE LAST (X,Y) POSITION
C     UPPERY IS THE Y OF THE BEGINNING (X,Y) POSITION
C     LOWERY IS THE Y OF THE LAST (X,Y) POSITION
C
C     ROW IS VARIABLE CONTAINING CURRENT ROW BEING PROCESSED
C     COLUMN IS VARIABLE CONTAINING CURRENT COLUMN BEING PROCESSED
C     STEP IS VARIABLE ALLOWING DIFFERENT SCAN METHODS
C
C     POINTS IS AN ARRAY WHICH IS PASSED TO AN ASSEMBLY LANGUAGE
C     SUBROUTINE WHICH DOES THE ACTUAL SCANNING.
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX,
     1 LEFTX,RIGHTX,UPPERY,LOWERY,STEP
      INTEGER ARRAY(100,100),AMAX(5,20)
      INTEGER LEFTX, RIGHTX, UPPERY, LOWERY
      INTEGER YDIS, XDIS, AREA, ROW, COLUMN
      INTEGER LOW, DELTAY, STEP, STEP2
      INTEGER POINTS(3, 500), SUM(256)
C
C
C     THIS PROGRAM CAN RUN ON EITHER TERMINAL FOR SCAN PURPOSES
C
C
C
C     SET UP BEGINNING LOCATION TO SCAN
C
      STEP2=2*STEP
      K = 0
      M = 0
      DELTAY = 0
      LOW = LOWERY
C
```

```
C
16        N=J+1
          IF(N.GT.100)GO TO 30
          IF(A(M,N).GE.LTH.AND.A(M,N).LE.UTH)GO TO 30
          IF(A(M,N).GE.LTHF)GO TO 30
25        IO=M
          JO=N
C
C         FOUND BEGINNING OF HOLE.  GO TRACE EDGE ETC.
C
          CALL HOLE(I,J,IO,JO,LTH,UTH,PERIM,NBIAS,B,NBP)
C
C         CHECK TO SEE IF POINTS FOUND.  DEPENDENT ON PERIMETER SIZE CLASSIFY
C         AS A HOLE OR SPOT.
          IF(PERIM.LE.0.0)GO TO 30
          IF(PERIM.LE.20.)NSPOT=NSPOT+1
          IF(PERIM.GT.20.)NHOLE=NHOLE+1
30        IF(A(I,J).GT.NBIAS)GO TO 146
C
C         FOUND INTERNAL POINT.  MARK IT WITH NBIAS.
C
          A(I,J)=NBIAS+A(I,J)
          IA=IA+1
          IXP=ISX+I*NXS
          IYP=ISY-J*NYS
          IF(IDEBUG.EQ.0)CALL SPDOT(IXP,IYP)
C
C         SET FLAG THAT WE ARE NOW INTERNAL TO SEGMENT.
          NFL=2
          GO TO 147
C
C         IF WE WERE ON EDGE OF SEGMENT AND CURRENT POINT HAS
C         NOT BEEN MARKED, THEN WE MUST BE MOVING COMPLETELY
C         OUTSIDE SEGMENT.  SET FLAG.
C
146       IF(A(I,J).LT.NBIAS.AND.NFL.EQ.1)NFL=0
          IF(A(I,J).GE.NBIAS)NFL=1
C
C         CALCULATE AREA OF SEGMENT.  METHOD USED INVOLVES LOOKING AT
C         CURRENT POINT AND THE THREE POINTS ADJACENT IN THE I-1, J-1
C         DIRECTION.  THESE 3 POINTS HAVE ALREADY BEEN EXAMINED FOR
C         BEING PART OF THE SEGMENT AND MARKED IF SO.  BY COUNTING THE
C         NUMBER OF CORNER POINTS THAT WERE IN THE SEGMENT, THEN
C         THE PORTION OF THE BOX THEY FORM THAT IS CONTAINED IN THE
C         SEGMENT CAN BE DETERMINED.  IF ALL 4 POINTS IN SEGMENT, THEN
C         WHOLE BOX CONTAINED.  IF 3 POINTS, THEN HALF BOX.  LESS THAN 3
C         POINTS, THEN THERE IS NO AREA ENCLOSED.  NOTE THAT THE
C         VALUES ADDED TO THE AREA COMPENSATE FOR THE NON-SQUARE SCAN.
C
147       NM=0
```

```
      IF(I.EQ. 1)GO TO 149
      IF(J.EQ. 1)GO TO 149
      IF(A(I,J).GE. NBIAS)NM=NM+1
      IF(A(I-1,J).GE. NBIAS)NM=NM+1
      IF(A(I-1,J-1).GE. NBIAS)NM=NM+1
      IF(A(I,J-1).GE. NBIAS)NM=NM+1
      IF(NM.EQ. 3)RAREA=RAREA+0. 625
      IF(NM.EQ. 4)RAREA=RAREA+1. 25
149   CONTINUE
150   CONTINUE
C
      RETURN
      END
```

```
C     PROGRAM DOTMAT
C
C     THIS PROGRAM PRODUCES A CARTOON OF THE IMAGE SCANNED USING
C     AN UPPER AND LOWER THRESHOLD VALUE  ENTERED THROUGH
C     THE CONSOLE.  BY SELECTING THE CORRECT THRESHOLDS ONE CAN
C     OBTAIN A RESONABLE PICTURE OF WHAT WAS SCANNED.
C     THE PROGRAM ALLOWS MULTIPLE PLOTS USING DIFFERENT THRESHOLDS
C     THUS ONE CAN FILL IN AREAS BY DETERMINING WHICH THRESHOLDS
C     WILL GENERATE WHICH CARTOON.  TO EXIT THE PROGRAM THRESHOLD
C     VALUES OF 0 FOR BOTH UPPER AND LOWER THRESHOLDS WILL CAUSE THE
C     PROGRAM TO GO TO STOP.  CHECKS ARE MADE DURING PROCESSING TO
C     MAKE SURE THE FILE IS VALID AND THE THRESHOLDS ARE CORRECT.
C     THE NUMBER OF POINTS WHICH WILL BE PLOTTED IS GIVEN ON THE
C     CONSOLE AFTER THE THRESHOLDS ARE TYPED IN.  THIS
C     TOTAL IS THEN PROGRAMATICALLY CALCULATED AND LATER CHECKED
C     AGAINST THE FIRST NUMBER TO VALIDATE PROCESSING.
C     SUBROUTINE LOGIC IS USED TO UNPACK DATA FROM BLOCKS 2 - 21.
C     SUBROUTINE SPDOT IS USED TO PLOT "." ON THE SCREEN OF THE
C     TEKTRONIX 4006-1 TERMINAL.
C
C
C     ARRAY WILL CONTAIN THE SCANNED VALUES OF EACH BLOCK WHICH
C     WILL BE UNPACKED BY SUBROUTINE LOGIC.
C     PACK WILL CONTAIN THE RELATIVE BLOCK 1 THRU RELATIVE
C     BLOCK 20 DATA WHICH WILL BE READ FROM THE FILENAME.
C     IHOLD WILL BE USED IN THE UNPACKING DO LOOP
C     SUM WILL HOLD THE HISTOGRAM DATA AND IS USED TO VALIDATE
C     THE FILENAME
C     TOT WILL BE USED TO ACCUMULATE TOTALS TO CHECK AGAINST
C     THE VALUES IN SUM TO MAKE SURE EVERYTHING IS OK
C
C     SIZE IS THE MULTIPLE OF A 100 X 100 SCAN FILE
C     WHICH WILL BE DISPLAYED ON THE SCREEN.  THUS A
C     SIZE OF 4 WILL GIVE A 400 X 400 CARTOON ON THE SCREEN.
C     UTH IS THE UPPER THRESHOLD TAKEN FROM THE CONSOLE
C     STEP IS THE WAY THE IMAGE WAS ORIGINALLY SCANNED BY SCAN2
C     AND IS NEEDED IN ORDER TO UNPACK THE DATA INTO ITS RELATIVE
C     SCANNED POSITION.
C     INAME IS THE NAME OF THE FILE TO BE PROCESSED.
C
C
C
      INTEGER ARRAY(500), PACK(256), IHOLD(256), SUM(256), TOT(256)
      INTEGER SIZE, UTH, UTHLOC, STEP
      DIMENSION INAME(6)
C
C     INITIALIZE ARRAY
C
      DO 1 I = 1, 500
      ARRAY(I) = 0
```

```
1       CONTINUE
C
C       ZERO OUT SEVERAL ARRAYS FOR INITIALIZATION PURPOSES
C
        DO 2 I = 1, 256
        PACK(I) = 0
        IHOLD(I) = 0
        SUM(I) = 0
        TOT(I) = 0
2       CONTINUE
C
C       ITOT SHOULD BE THE TOTAL NUMBER OF POINTS IN SUM
C       ICOUNT IS COUNTER INCREMENTED EACH TIME A DOT IS DISPLAYED
C       IBLOCK IS CURRENT BLOCK BEING PROCESSED AND IS COMPARED
C       AGAINST IBLK WHICH IS EQUAL TO PACK(1) OF THE BLOCK JUST
C       READ. THIS COMPARISON IS DONE FOR VALIDATION PURPOSES.
C       IEND IS FLAG SET AFTER BLOCK 20 IS READ TO PREVENT READING
C       PAST VALID DATA. IEND IS SET TO 1 AFTER BLOCK 20 IS READ.
        ITOT = 0
        ICOUNT = 0
        IBLOCK = 0
        IEND = 0
C
C       THIS PROGRAM SHOULD BE RUN FROM THE DGC TERMINAL.
C
        CALL OPEN(1, "$TTO1",0,IER,128)
        TYPE "  ENTER FILENAME UP TO 10 CHARACTERS :"
        READ(11, 100) INAME(1)
100     FORMAT (S10)
        WRITE (10, 101) INAME(1)
101     FORMAT ("  FILENAME IS : ", S10)
        CALL FOPEN(2, INAME, 512)
        TYPE "  FILE IS OPEN"
C
C       LOCATE TO RELATIVE BLOCK 0 FOR HISTOGRAM INFO
C
        CALL FSEEK(2, 0)
C
C       READ RELATIVE BLOCK 0 INTO SUM
C
        READ BINARY(2) SUM
C
C       CHECK TO SEE IF 10000 POINTS IN BLOCK 0
C
        DO 3 I = 1, 256
        ITOT = ITOT + SUM(I)
3       CONTINUE
        IF (ITOT.NE.10000) GO TO 990
C
C       INITIALIZE VARIABLES FOR MULTIPLE PASSES
```

```
C       THESE VARIABLES WILL BE REINITIALIZED FOR EACH 2 NEW
C       UPPER AND LOWER THRESHOLDS ENTERED THROUGH THE CONSOLE.
C       LTH IS LOWER THRESHOLD
C       ITOTPTS IS TOTAL NUMBER OF POINTS FROM LOWER TO UPPER
C       THRESHOLD WHICH SHOULD BE PLOTTED BASED ON THE VALUES
C       FROM THE ARRAY CALLED SUM.  THIS TOTAL (ITOTPTS) IS CHECKED
C       AGAINST ICOUNT TO VERIFY IF EVERYTHING IS OFK.
C
31      UTH = 0
        LTH = 0
        UTHLOC = 0
        LTHLOC = 0
        ITOTPTS = 0
        SIZE = 0
        STEP = 2
C
C
C
        ACCEPT "  ENTER LOWER THRESHOLD :  ", LTH
        ACCEPT "  ENTER UPPER THRESHOLD :  ", UTH
C
C       VERIFY THE THRESHOLDS ACCEPTED FROM CONSOLE
C
        IF (UTH.LT.LTH) GO TO 991
        IF (LTH.EQ.0.AND.UTH.EQ.0) GO TO 999
        IF (LTH.LT.0.OR.UTH.GT.255) GO TO 992
C
C       CALCULATE NUMBER OF POINTS THAT SHOULD PRINT
C       USING BLOCK 0 MATRIX
C
        LTHLOC = LTH + 1
        UTHLOC = UTH + 1
        DO 4 M = LTHLOC, UTHLOC
        ITOTPTS = ITOTPTS + SUM(M)
4       CONTINUE
      TOTL=ITOTPTS
        TYPE "  TOTAL NO OF POINTS TO PLOT IS :  ", TOT
      ACCEPT "ENTER STANDARD NUMBER OF POINTS FOR CARD BEING TESTED",STNRD
      CALL TEST(TOTL,STNRD)
        IF (ITOTPTS.EQ.0)  GO TO 993
C
C       INITIALIZE  DISK FILE TO READ BLOCKS
C
        SIZE = 4
        IMIDX = 512
        IMIDY = 370
        ISTARTX = IMIDX - ((SIZE * 100) / 2)
        ISTARTY = IMIDY + ((SIZE * 100) / 2)
        IULX = ISTARTX
        IULY = ISTARTY
```

```
          IBLOCK = 0
          IEND = 0
          IBLK = 0
          ICOUNT = 0
C
C         READ BLOCK INTO ARRAY CALLED PACK
C
5         READ BINARY(2) PACK
          IBLOCK = IBLOCK + 1
          IBLK = PACK(1)
          IF (IBLOCK.NE.IBLK) TYPE "  BLOCK COUNTS NOT EQUAL - BUG"
          IF (IBLK.EQ.20) IEND = 1
C
C         UNPACK DATA
C
          J = 6
          DO 6 I = 1, 500, 2
          J = J + 1
          ARRAY(I) = IRIGHT(PACK(J),8)
          IHOLD(J) = ILEFT(ARRAY(I),8)
          ARRAY(I + 1) = PACK(J) - IHOLD(J)
6         CONTINUE
C
C         SEARCH ARRAY TO DETERMINE WHICH POINTS TO PLOT BASED ON
C         WHETHER THE VALUE LIES BETWEEN THE LOWER AND UPPER
C         THRESHOLDS.
C
C         TWO PASSES WILL BE MADE DOWN EACH COLUMN OF THE SCREEN
C         BECAUSE THE SCAN2 PROGRAM SCANNED IN THIS MANNER.
C
C
          J = 0
          K = 0
          DO 10 I = 1, 500
          IF (ARRAY(I).GE.LTH.AND.ARRAY(I).LE.UTH) GO TO 11
12        J = J + 1
          IF (J.EQ.50) GO TO 13
          GO TO 10
13        K = K + 1
          GO TO (14, 15, 14, 15, 14, 15, 14, 15, 14, 15), K
14        J = 0
          IULY = ISTARTY - SIZE
          GO TO 10
15        J = 0
          IULY = ISTARTY
          IULX = IULX + SIZE
          GO TO 10
C
C         PRINT DOT FOR THRESHOLD VALUE FOUND
C
```

```
11      IYPOS = IULY - (STEP*SIZE)*J
        IXPOS = IULX
        CALL SPDOT (IXPOS, IYPOS)
        ICOUNT = ICOUNT + 1
        GO TO 12
C
C
C
10      CONTINUE
C
C
C
        IULY = ISTARTY
        IF (IEND.EQ.1) GO TO 99
        GO TO 5
C
C
99      IF (ICOUNT.NE.ITOTPTS) GO TO 994
        CALL FSEEK(2, 1)
        GO TO 31
990     TYPE "  ITOT NE 10000 - BAD DISK "
        GO TO 999
991     TYPE "  UPPER THRESHOLD LT LOWER THRESHOLD - TRY AGAIN"
        GO TO 31
992     TYPE "  THRESHOLD VALUES OUT OF RANGE 0-255 TRY AGAIN"
        GO TO 31
993     TYPE "  NUMBER OF POINTS TO PLOT IS ZERO - TRY AGAIN"
        GO TO 31
994     TYPE "  ICOUNT NOT EQUAL TO ITOTPTS - PROGRAM BUG"
C
C
C
999     CALL RESET
        STOP
        END
```

```
C       PRTMAT
C               THIS PROGRAM PRINTS THE IMAGE ON THE LINE PRINTER.
C       THE THRESHOLDING AND OTHER DATA MANIPULATION IS SIMILAR TO
C       THAT FOUND IN THE PROGRAM 'DOTMAT', WHICH IS COMMENTED IN GREATER
C       DETAIL.  THE PRIMARY DIFFERENCE IS THAT THIS PROGRAM PRINTS
C       THE IMAGE SIDEWAYS, TO PREVENT IMAGE REVERSAL.  OF COURSE A "#"
C       IS PRINTED INSTEAD OF A DOT, AND THE BLANK SPACES ARE CHARACTER
C       SPACES.
C
C       THIS PROGRAM PRODUCES A CARTOON OF THE IMAGE SCANNED USING
C       AN UPPER AND LOWER THRESHOLD VALUE ENTERED THROUGH
C       THE CONSOLE.  BY SELECTING THE CORRECT THRESHOLDS ONE CAN
C       OBTAIN A RESONABLE PICTURE OF WHAT WAS SCANNED.
C       TO EXIT THE PROGRAM THRESHOLD VALUES OF 0 FOR BOTH UPPER
C       AND LOWER THRESHOLDS WILL CAUSE THE PROGRAM TO STOP.
C       CHECKS ARE MADE DURING PROCESSING TO MAKE SURE THE FILE
C       IS VALID AND THE THRESHOLDS ARE CORRECT.
C       THE NUMBER OF POINTS WHICH WILL BE PLOTTED IS GIVEN ON THE
C       CONSOLE AFTER THE THRESHOLDS ARE TYPED IN.  THIS TOTAL
C       IS THEN PROGRAMATICALLY CALCULATED AND LATER CHECKED
C       AGAINST THE FIRST NUMBER TO VALIDATE PROCESSING.
C       SUBROUTINE LOGIC IS USED TO UNPACK DATA FROM BLOCKS 2-21.
C
C       ARRAY WILL CONTAIN THE SCANNED VALUES OF EACH BLOCK WHICH
C       WILL BE UNPACKED BY SUBROUTINE LOGIC.
C       PACK WILL CONTAIN THE RELATIVE BLOCK 1 THRU RELATIVE
C       BLOCK 20 DATA WHICH WILL BE READ FROM THE FILENAME.
C       IHOLD WILL BE USED IN THE UNPACKING DO LOOP
C       SUM WILL HOLD THE HISTOGRAM DATA AND IS USED TO VALIDATE
C       THE FILENAME.
C       TOT WILL BE USED TO ACCUMULATE TOTALS TO CHECK AGAINST
C       THE VALUES IN SUM TO MAKE SURE EVERYTHING IS OK
C
C       UTH IS THE UPPER THRESHOLD TAKEN FROM THE CONSOLE.
C       STEP IS THE WAY THE IMAGE WAS ORIGINALLY SCANNED BY SCAN2
C       AND IS NEEDED IN ORDER TO UNPACK THE DATA INTO ITS RELATIVE
C       SCANNED POSITION.
C       INAME IS THE NAME OF THE FILE TO BE PROCESSED.
C
C
C
C
        INTEGER ARRAY(500), PACK(256), IHOLD(256), SUM(256), TOT(256)
        INTEGER SIZE, UTH, UTHLOC, STEP
        INTEGER PRTLINE(100)
        DIMENSION INAME(6)
C
C       INITIALIZE ARRAY
C
        DO 1 I = 1, 500
        ARRAY(I) = 0
1       CONTINUE
C
C       ZERO OUT SEVERAL ARRAYS FOR INITIALIZATION PURPOSES
C
        DO 2 I = 1, 256
        PACK(I) = 0
        IHOLD(I) = 0
        SUM(I) = 0
        TOT(I) = 0
2       CONTINUE
        DO 21 I = 1, 100
        PRTLINE(I) = " "
21      CONTINUE
C
C       ITOT SHOULD BE THE TOTAL NUMBER OF POINTS IN SUM
C       ICOUNT IS COUNTER INCREMENTED EACH TIME A POINT IS PLOTTED.
C       IBLOCK IS CURRENT BLOCK BEING PROCESSED AND IS COMPARED
C       AGAINST IBLK WHICH IS EQUAL TO PACK(1) OF THE BLOCK JUST
C       READ.  THIS COMPARISON IS DONE FOR VALIDATION PURPOSES.
C       IEND IS FLAG SET AFTER BLOCK 20 IS READ TO PREVENT READING
C       PAST VALID DATA.  IEND IS SET TO 1 AFTER BLOCK 20 IS READ.
C
        ITOT = 0
        ICOUNT = 0
        IBLOCK = 0
        IEND = 0
C
C       THIS PROGRAM CAN RUN ON EITHER TERMINAL
C
        CALL FOPEN(12, "$LPT")
        TYPE "  ENTER FILENAME UP TO 10 CHARACTERS :"
        READ(11, 100) INAME(1)
100     FORMAT (S10)
        WRITE (10, 101) INAME(1)
101     FORMAT ("  FILENAME IS : ", S10)
        CALL FOPEN(2, INAME, 512)
        TYPE "  FILE IS OPEN"
C
C       LOCATE TO RELATIVE BLOCK 0 FOR HISTOGRAM INFO
C
        CALL FSEEK(2, 0)
C
C       READ BLOCK 0 INTO ARRAY CALLED SUM
C
        READ BINARY(2) SUM
C
C       CHECK TO SEE IF 10000 POINTS IN BLOCK 0
C
        DO 3 I = 1, 256
```

```
         ITOT = ITOT + SUM(I)
3        CONTINUE
         IF (ITOT.NE.10000) GO TO 990
C
C        INITIALIZE VARIABLES FOR MULTIPLE PASSES
C        THESE VARIABLES WILL BE REINITIALIZED FOR EACH 2 NEW
C        UPPER AND LOWER THRESHOLDS ENTERED THROUGH THE CONSOLE.
C        LTH IS LOWER THRESHOLD.
C        ITOTPTS IS TOTAL NUMBER OF POINTS FROM LOWER TO UPPER
C        THRESHOLD WHICH SHOULD BE PLOTTED BASED ON THE VALUES
C        FROM THE ARRAY CALLED SUM.  THIS TOTAL (ITOTPTS) IS CHECKED
C        AGAINST ICOUNT TO VERIFY IS EVERYTHING IS OK.
C
C
31       UTH = 0
         LTH = 0
         UTHLOC = 0
         LTHLOC = 0
         ITOTPTS = 0
         SIZE = 0
         STEP = 2
C
C
C
         ACCEPT "  ENTER LOWER THRESHOLD :  ", LTH
         ACCEPT "  ENTER UPPER THRESHOLD :  ", UTH
C
C        VERIFY THE THRESHOLDS ACCEPTED FROM CONSOLE
C
         IF (UTH.LT.LTH) GO TO 991
         IF (LTH.EQ.0.AND.UTH.EQ.0) GO TO 999
         IF (LTH.LT.0.OR.UTH.GT.255) GO TO 992
C
C        CALCULATE NUMBER OF POINTS THAT SHOULD PRINT
C        USING BLOCK 0 MATRIX
C
         LTHLOC = LTH + 1
         UTHLOC = UTH + 1
         DO 4 M = LTHLOC, UTHLOC
         ITOTPTS = ITOTPTS + SUM(M)
4        CONTINUE
         TOTL=ITOTPTS
         TYPE "  TOTAL NO OF POINTS TO PLOT IS :  ", ITOTPTS
         ACCEPT"ENTER STANDARD NUMBER OF POINTS TO PLOT IS>",STNRD
         CALL TEST(TOTL,STNRD)
         IF (ITOTPTS.EQ.0)  GO TO 993
C
C        SET UP VARIABLES TO PRINT
C
         L = 100
```

```
      DO 32 I = 1, 100
      PRTLINE(I) = " "
32    CONTINUE
C
C     WRITE MESSAGE TO GIVE FILENAME AND OTHER INFO
C
      WRITE (12, 102) INAME(1), LTH, UTH, ITOTPTS
102   FORMAT ("<14>", "  FILENAME IS : ", S10, "  LOWER THRESHOLD IS : ",
     2I3, "  UPPER THRESHOLD IS : ", I3, "  TOTAL POINTS TO BE PLOTTED IS :
     3I5)
C
C     INITIALIZE  DISK FILE TO READ BLOCKS
C
      SIZE = 4
      IMIDX = 512
      IMIDY = 370
      ISTARTX = IMIDX - ((SIZE * 100) / 2)
      ISTARTY = IMIDY + ((SIZE * 100) / 2)
      IULX = ISTARTX
      IULY = ISTARTY
      IBLOCK = 0
      IEND = 0
      IBLK = 0
      ICOUNT = 0
C
C     READ BLOCK INTO ARRAY CALLED PACK
C
5     READ BINARY(2) PACK
      IBLOCK = IBLOCK + 1
      IBLK = PACK(1)
      IF (IBLOCK.NE.IBLK) TYPE "  BLOCK COUNTS NOT EQUAL - BUG"
      IF (IBLK.EQ.20) IEND = 1
C
C     UNPACK DATA
C
      J = 6
      DO 6 I = 1, 500, 2
      J = J + 1
      ARRAY(I) = IRIGHT(PACK(J),8)
      IHOLD(J) = ILEFT(ARRAY(I),8)
      ARRAY(I + 1) = PACK(J) - IHOLD(J)
6     CONTINUE
C
C
C
      L = 100
      J = 0
      K = 0
      DO 10 I = 1, 500
      IF (ARRAY(I).GE.LTH.AND.ARRAY(I).LE.UTH) GO TO 11
```

```
12        J = J + 1
          L = L - 2
          IF (J. EQ. 50) GO TO 13
          GO TO 10
13        K = K + 1
          GO TO (14, 15, 14, 15, 14, 15, 14, 15, 14, 15), K
14        J = 0
          IULY = ISTARTY - SIZE
          L = 99
          GO TO 10
15        J = 0
          IULY = ISTARTY
          IULX = IULX + SIZE
          WRITE (12, 200) (PRTLINE(N), N = 1, 100)
200       FORMAT (5X, 100A1)
          DO 16 N = 1, 100
          PRTLINE(N) = " "
16        CONTINUE
          L = 100
          GO TO 10
C
C
11        IYPOS = IULY - (STEP*SIZE)*J
          IXPOS = IULX
          IXEND = IULX
          IYEND = IYPOS
          ICOUNT = ICOUNT + 1
          PRTLINE(L) = "#"
          GO TO 12
C
C
C
10        CONTINUE
C
C
C
          IULY = ISTARTY
          IF (IEND. EQ. 1) GO TO 99
          GO TO 5
C
C
99        IF (ICOUNT. NE. ITOTPTS) GO TO 994
C
C         LOCATE BACK TO RELATIVE BLOCK 1 FOR MULTIPLE PASSES
C
          CALL FSEEK(2, 1)
          GO TO 31
C
C         ERROR MESSAGES
C
```

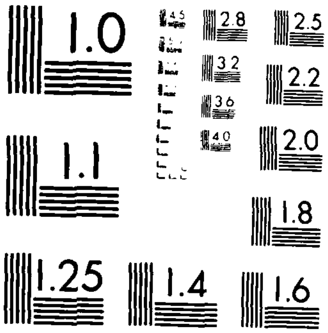
```
990     TYPE "  ITOT NE 10000 - BAD DISK "
        GO TO 999
991     TYPE "  UPPER THRESHOLD LT LOWER THRESHOLD - TRY AGAIN"
        GO TO 31
992     TYPE "  THRESHOLD VALUES OUT OF RANGE 0-255 TRY AGAIN"
        GO TO 31
993     TYPE "  NUMBER OF POINTS TO PLOT IS ZERO - TRY AGAIN"
        GO TO 31
994     TYPE "  ICOUNT NOT EQUAL TO ITOTPTS - PROGRAM BUG"
C
C
C
999     CALL RESET
        STOP
        END


        SUBROUTINE TEST(POINTS,STNRD)
        IF (ABS(POINTS-STNRD)/STNRD-.005)1,1,2
      1 WRITE(12,50)
        GO TO 88
      2 WRITE(12,51)
     50 FORMAT(1X,8HBOARD OK)
     51 FORMAT(1X,23HPROBABLE ERROR ON BOARD)
     88 RETURN
        END
```

```
C        PROGRAM HISTOGRAM
C
C        THIS PROGRAM DISPLAYS A HISTOGRAM OF THE SCAN FILES
C        CREATED BY PROGRAM SCAN2.  THE TEKTRONIX 4006-1
C        TERMINAL IS USED FOR DISPLAY.   BOTH TERMINALS ARE NEEDED TO
C        RUN THIS PROGRAM.   THE PROGRAM SHOULD BE RUN ON THE DGC
C        TERMINAL WITH OUTPUT OF THE HISTOGRAM GOING TO THE TEKTRONIX
C        TERMINAL.   THE NAME OF THE FILE WHOSE HISTOGRAM IS WANTED IS
C        ENTERED.   THE PROGRAM ONLY HAS TO ACCESS THE
C        FIRST BLOCK OF THE FILE TO OBTAIN THE HISTOGRAM DATA.   THIS
C        DATA IS READ INTO AN INTEGER ARRAY CALLED SUM WHICH IS
C        256 WORDS IN LENGTH.   A DO LOOP THROUGH THIS ARRAY CREATES
C        A SERIES OF LINES WHOSE LENGTH EQUALS THE VALUE IN THE
C        WORD BEING PROCESSED.   A SUBROUTINE CALL TO SPLIN PLOTS A
C        HORIZONTAL LINE ON THE SCREEN.   256 LINES WILL BE PLOTTED.
C        AFTER THE LINES ARE PLOTTED, THE SUBROUTINES SPLIN AND SPMOV
C        ARE USED TO PRODUCE SCALE VALUES UNDER THE HISTOGRAM USEFUL
C        IN OBTAINING THRESHOLD VALUES FOR VALLEYS AND PEAKS.
C
C
C        SUM CONTAINS THE FIRST BLOCK OF DATA READ FROM THE INPUT FILE
C        INAME IS NAME OF FILE TO BE READ
C
         INTEGER SUM(256)
         DIMENSION INAME(6)
C
C        INITIALIZE SUM
C
         DO 1 I = 1, 256
         SUM(I) = 0
1        CONTINUE
         TYPE "  ENTER FILENAME UP TO 10 CHARACTERS : "
         READ (11, 100) INAME(1)
100      FORMAT (S10)
         WRITE (10, 101) INAME(1)
101      FORMAT ("  FILENAME IS : ", S10)
         CALL FOPEN(2, INAME, 512)
         TYPE "   FILE IS OPEN  "
         CALL OPEN (1, "$TT01",0,IER,128)
C
C        SET UP INITIAL STARTING POSITION FOR PLOTTING
C
         IXPOS = 2
         IYPOS = 60
         IXEND = 2
         IYEND = 60
C
C        READ FIRST BLOCK OF DATA FROM FILE OPENED INTO ARRAY SUM
C
         READ BINARY(2) SUM
```

AD-A151 976 AUTOMATIC EVALUATION OF PRINTED WIRING BOARD SOLDER(U) 2/2
SCI SYSTEMS INC HUNTSVILLE AL 04 MAY 79
DAAK40-77-C-0105

UNCLASSIFIED F/G 9/5 NL

END
FILMED
DTIC


1.0
4.5
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
C
C       CHECK DISK FILE OUT TO SEE IF CREATED BY SCAN2
C
        ITOT = 0
        DO 11 J = 1, 256
        ITOT = ITOT + SUM(J)
11      CONTINUE
C
C       IF ITOT IS NOT EQUAL TO 10000 THE DISK FILE IS NOT GOOD
C
        IF (ITOT.EQ.10000) GO TO 13
        TYPE "   FILENAME WAS NOT CREATED BY SCAN2 JOB ENDS"
        GO TO 99
C
C
C
C       PLOT 256 HORIZONTAL LINES
C
13      DO 2 I = 1, 256
        IYEND = IYEND + SUM(I)
        CALL SPLIN (IXPOS, IYPOS, IXEND, IYEND)
        IXEND = IXEND + 4
        IXPOS = IXEND
        IYEND = 60
2       CONTINUE
C
C       SET UP TO PRINT NUMBERS UNDER HISTOGRAM
C
C
C       GET BACK TO STARTING POINT
C
        IXPOS = 2
        IYPOS = 60
        IXEND = 2
        IYEND = 60
        CALL SPLIN (IXPOS, IYPOS, IXEND, IYEND)
C
C       DO THREE LONG LINES
C
        IYEND = 40
        DO 20 J = 1, 3
        CALL SPLIN(IXPOS, IYPOS, IXEND, IYEND)
        IXPOS = IXPOS + 400
        IXEND = IXPOS
20      CONTINUE
C
C       RESET VARIABLES AND DO THREE MIDDLE LINES
C
        IYEND = 50
        IXPOS = 202
        IXEND = 202
        DO 30 J = 1, 3
        CALL SPLIN (IXPOS, IYPOS, IXEND, IYEND)
        IXPOS = IXPOS + 400
        IXEND = IXPOS
30      CONTINUE
C
C       SET UP VARIABLES FOR 26 SMALL LINES
C
        IYEND = 55
        IYPOS = 60
        IXPOS = 2
        IXEND = 2
        DO 40 J = 1, 26
        CALL SPLIN (IXPOS, IYPOS, IXEND, IYEND)
        IXPOS = IXPOS + 40
        IXEND = IXPOS
40      CONTINUE
C
C
C
C       GET BACK TO PRINT NUMBERS UNDER HISTOGRAM
C
        IXPOS = 2
        IYPOS = 20
        IXEND = 2
        IYEND = 20
        CALL SPMOV (IXPOS, IYPOS)
        WRITE (1) "0"
C
C       PRINT NUMBER 50
C
        IXPOS = 192
        IYPOS = 20
        IXEND = 192
        IYEND = 20
        CALL SPMOV(IXPOS, IYPOS)
        WRITE(1) "50"
C
C       DO LOOP TO GIVE 100, 150, 200, 250
C
        IXPOS = 385
        IYPOS = 20
        IXEND = 385
        IYEND = 20
C
C
        DO 50 I = 1, 4
        CALL SPMOV(IXPOS, IYPOS)
        GO TO (51, 52, 53, 54), I
```

```
51      WRITE(1) "100"
        GO TO 55
52      WRITE(1) "150"
        GO TO 55
53      WRITE(1) "200"
        GO TO 55
54      WRITE(1) "250"
        GO TO 55
55      IXPOS = IXPOS + 200
        IXEND = IXPOS
50      CONTINUE
99      CALL RESET
        STOP
        END
```

```
      SUBROUTINE TRACE(IROW,JCOL,IO,JO,LTH,UTH,PERIM,AREA,NSPL,
     1 IMIN,IMAX,JMIN,JMAX,B,NPT)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C     ONCE AN AREA, OR HOLE WITHIN AN AREA, HAS BEEN LOCATED
C     THIS SUBROUTINE TRACES THE EDGES AND MARKS THEM AS PART
C     OF THE AREA  ADDITIONALLY, TRACE BUILDS A TWO DIMENSIONAL ARRAY, B,
C     WHICH CONTAINS THE ORDERED I AND J VALUES OF THE EDGE POINTS OF
C     THE AREA OF HOLE BEING TRACED.
C
C
C     SOME OF THE VARIABLES USED ARE:
C
C     IROW,JCOL           STARTING COORDINATES, WITHIN ARRAY, OF THE EDGE TO
C                         BE TRACED.
C     IO,JO               NEXT TO LAST POINT FOUND ON EDGE BEFORE CURRENT POINT.
C     IP,JP               LAST POINT FOUND, USED AS CENTER OF SEARCH FOR NEXT PT
      LTH,UTH             LOWER AND UPPER THRESHOLDS PASSED IN FROM BOARD.  USED
C                         AS DELIMITERS FOR THE FEATURES IN THE SCAN.
C     AREA                NUMBER USED TO MARK POINTS AS BELONGING TO A GROUP.
C     NSPL                NUMBER USED TO MARK POINTS, WITHIN AN AREA, AS
C                         BEING PART OF A SUBGROUP( EDGE OF HOLE).
C             **          BOTH AREA AND NSPL ARE POWERS OF 2 SO THEY BASICLLY
C                         JUST SET A BIT.
C
C     IMIN,IMAX           THE MIN AND MAX COLUMNS OF THE EDGE.
C     JMIN,JMAX           THE MIN AND MAX ROWS OF THE EDGE.
C
C     PERIM               MEASURED PERIMETER OF THE AREA TRACED AROUND.
C
C     B                   AN ARRAY CONTAINING THE I,J VALUES OF THE EDGE.
C     NPT                 NUMBER OF POINTS IN B WHICH ARE THE CURRENT EDGE.
C
C
      COMMON ARRAY,SUM,LP,IMN,IMX,JMN,JMX,NCNT,NAREA,AMAX(5,20)
      COMMON /PLT/ISTARTX,ISTARTY,MIDX,MIDY,NXSIZE,NYSIZE
      INTEGER ARRAY(100,100), SUM(256)
      INTEGER AREA, UTH,UTHF
      INTEGER B(2,1500)
C
C     VERIFY STARTING ROW AND COLUMN ARE VALID.
C
      IF(IROW.GT.100)GO TO 905
      IF(IROW.LT.1)GO TO 905
      IF(JCOL.GT.100)GO TO 905
      IF(JCOL.LT.1)GO TO 905
      I=IROW
      J=JCOL
```

```
C
C     SET STARTING POINT AS FIRST IN EDGE
C
      NPT=1
      B(1,NPT)=IROW
      B(2,NPT)=JCOL
C
C     INITIALIZE PERIMETER AND MIN MAX
C
      PERIM=0.0
      IMAX=IROW
      IMIN=IROW
      JMAX=JCOL
      JMIN=JCOL
      IXE=0
C
C     ONCE A POINT HAS BEEN MARKED BY ADDING AREA TO IT, THEN
C     YOU NEED A DIFFERENT THRESHOLD VALUE. SET LOWER AND UPPER
C     THRESHOLDS SO WE CAN STILL IDENTIFY POINTS FROM THE CURRENT
C     EDGE AFTER THEY HAVE BEEN "MARKED".
C
      LTHF=LTH+AREA
      UTHF=UTH+AREA+NSPL
C
C     DI, DJ, DIAG ARE THE VALUES USED IN CALCULATING THE PERIMETER
C     THEY VARY DEPENDENT ON DIRECTION OF MOVEMENT FROM ONE POINT
C     TO THE NEXT. THE DIFFERENCE IN DI,DJ IS DUE TO THE CAMERA
C     NOT HAVING A SQUARE SCAN.
C
      DI=1.25
      DJ=1.0
      DIAG=1.60078
C
C     INITIALIZE VALUES FOR TEKTRONIX PLOTTING.
C
      IXP=ISTARTX+I*NXSIZE
      IYP=ISTARTY-J*NYSIZE
      MS=2
      GO TO 21
20    CONTINUE
C
C     HAVE WE GONE FULL CIRCLE
C
      IF(I.EQ.IROW.AND.J.EQ.JCOL)GO TO 77
C
C
C
C     PUT AREA FLAG ON DATA POINT
C
      ARRAY(I,J)=AREA+NSPL+ARRAY(I,J)
```

```
C
C        IS NEW POINT A MIN OR MAX VALUE
C
         IF(I.LT.IMIN)IMIN=I
         IF(I.GT.IMAX)IMAX=I
         IF(J.LT.JMIN)JMIN=J
         IF(J.GT.JMAX)JMAX=J
C
C        IF MARKING NEW EDGE, PLOT IT.
C
         IF(AREA.EQ.0)GO TO 10
C
C        NS IS A VALUE, 1 TO 8, WHICH IS THE DIRECTION THE EDGE IS MOVING.
C        AS LONG AS POINTS CONTINUE IN SAME DIRECTION, SKIP PLOT AND THEN
C        DRAW ONE LONG VECTOR INSTEAD.
C
         IF(MS.EQ.NS)GO TO 10
         CALL SPLIN(IXP,IYP,IXE,IYE)
         IXP=IXE
         IYP=IYE
         MS=NS
10       CONTINUE
C
C        ADD CORRECT VALUE TO PERIMETER.
C
         IF(I.EQ.IP)GO TO 11
         IF(J.EQ.JP)GO TO 12
         PERIM=PERIM+DIAG
         GO TO 13
11       PERIM=PERIM+DJ
         GO TO 13
12       PERIM=PERIM+DI
13       CONTINUE
C
C        PLACE NEW POINT IN ARRAY B.
C
         NPT=NPT+1
         IF(NPT.GT.1500)GO TO 900
         B(1,NPT)=I
         B(2,NPT)=J
21       CONTINUE
C
C        SAVE CURRENT POINT FOR PLOTTING.
C
         IXE=ISTARTX+I*NXSIZE
         IYE=ISTARTY-J*NYSIZE
         IP=I
         JP=J
C
C        FOUND AND PLOTTED A POINT.  SEARCH FOR NEXT POINT BEGINS.
```

```
C       ASSUME THE SEARCH OF ADJACENT POINTS,THE 8 POINTS TO
C       BE EXAMINED ARE NUMBERED 1 TO 8 STARTING WITH THE ONE
C       AT THE I-1,J POSITION AS 1 AND MOVING
C       CLOCKWISE. FOR THE POINT WE ARE CURRENTLY AT, THERE
C       WILL BE A PREVIOUS POINT AS ONE OF THE 8  OUR
C       EXAMINATION WILL START WITH THE NEXT CLOCKWISE POINT
C       AND PROCEED.
C
        IF(J-JO) 125, 120, 115
115     IF(I-IO) 118, 117, 116
116     NS=3
        GO TO 18
117     NS=4
        GO TO 18
118     NS=5
        GO TO 18
120     NS=2
        IF(I.LT.IO)NS=6
        GO TO 18
125     IF(I-IO) 126, 127, 128
126     NS=7
        GO TO 18
127     NS=8
        GO TO 18
128     NS=1
C
C
C       THROUGH WITH OLD POINTS.  SAVE CURRENT POINTS FOR USE NEXT TIME
C
18      CONTINUE
        IO=I
        JO=J
        GO TO (25,30,35,40,45,50,55,60),NS
C
C       NOW WE KNOW WHERE TO START, CIRCLE CLOCKWISE LOOKING FOR
C       NEXT POINT IN AREA BOUNDARY.
C
25      CONTINUE
C
C       TO PREVENT LOOPING FOR SPECIAL CASE OF ISOLATED POINT
C       ON ROW 1 (I=1), NEED FOLLOWING CHECK.
        IF(IP.EQ.1.AND.NS.EQ.2)GO TO 65
C
        I=IP-1
        IF(I.LT.1)GO TO 34
        J=JP
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.2)GO TO 65
```

```
C
C
C
30      CONTINUE
        I=IP-1
        IF(I.LT.1) GO TO 35
        J=JP-1
        IF(J.LT.1)GO  TO 45
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.3)GO TO 65
C
C
C
34      CONTINUE
C
C       DITTO COMMENT AT 25
        IF(IP.EQ.1.AND.NS.EQ.2)GO TO 65
C
35      CONTINUE
        J=JP-1
        IF(J.LT.1)GO TO 45
        I=IP
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.4)GO TO 65
C
C
C
40      I=IP+1
        IF(I.GT.100)GO TO 55
        J=JP-1
        IF(J.LT.1)GO TO 45
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.5) GO TO 65
C
C
C
45      I=IP+1
        IF(I.GT.100)GO TO 55
        J=JP
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.6)GO TO 65
C
```

```
C
C
50      I=IP+1
        IF(I.GT 100)GO TO 55
        J=JP+1
        IF(J.GT.100)GO TO 25
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.7)GO TO 65
C
C
C
55      J=JP+1
        IF(J.GT.100)GO TO 25
        I=IP
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.8)GO TO 65
C
C
C
60      I=IP-1
        IF(I.LT.1)GO TO 34
        J=JP+1
        IF(J.GT.100)GO TO 25
        IV=ARRAY(I,J)
        IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
        IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
        IF(NS.EQ.1) GO TO 65
        GO TO 25
C
C
C       COMPLETED CIRCLE WITH NO FIND.  SEE IF WE ARE STILL ON
C       FIRST POINT IDENTIFIED.
65      CONTINUE
        IF(IP.EQ.IROW.AND.JP.EQ.JCOL)GO TO 66
        TYPE "BOY ARE WE LOST!"
        GO TO 78
C
C
C
C
C       STILL ON FIRST POINT SO CONSIDER IT AS ISOLATED BY
C       SETTING VALUE TO ZERO
66      ARRAY(IROW,JCOL)=0
        PERIM=0.0
        GO TO 78
C
```

```
C       END OF LOOP  FLAG STARTING POINT AND CONTINUE
C
76      CONTINUE
        IF(I.NE.IROW.OR.J.NE.JCOL) GO TO 10
77      ARRAY(IROW,JCOL)=AREA+NSPL+ARRAY(IROW,JCOL)
        NPT=NPT+1
        IF(NPT.GT.1500)GO TO 900
        B(1,NPT)=I
        B(2,NPT)=J
        IF(I.EQ.IP)GO TO 175
        IF(J.EQ.JP)GO TO 176
        PERIM=PERIM+DIAG
        GO TO 177
175     PERIM=PERIM+DJ
        GO TO 177
176     PERIM=PERIM+DI
177     CONTINUE
        IXE=ISTARTX+IROW*NXSIZE
        IYE=ISTARTY-JCOL*NYSIZE
        IF(AREA.NE.0)CALL SPLIN(IXP,IYP,IXE,IYE)
C
C
78      CONTINUE
        RETURN
900     TYPE " TRACE FOUND > 1500 POINTS IN ONE AREA."
        STOP
 905    TYPE " BAD TRACE-",IROW,JCOL
        STOP
        END
```

```
      SUBROUTINE TRACK(IROW,JCOL,IO,JO,LT1,UT1,ZMIN,MTOUCH)
C
      PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C     THIS SUBROUTINES DRAWS A BORDER OF WIDTH ZMIN AROUND A SEGMENT
C     BOUNDARY  IF ANY OTHER SEGMENTS ARE ENCOUNTERED WITHIN THE BORDER,
C     THEN THE FLAG, MTOUCH, IS RETURNED INDICATING CRITICAL CLOSENESS
C     TRACK USES THE SAME ALGORITHM AS TRACE FOR FINDING THE EDGE POINTS
C     OF A SEGMENT WHOSE POINTS ARE BETWEEN THE VALUES LT1, UT1.
C
C
C
      COMMON ARRAY,SUM,LP,IMN,IMX,JMN,JMX,NCNT,NAREA,AMAX
      COMMON /PLT/ISTARTX,ISTARTY,MIDX,MIDY,NXSIZE,NYSIZE
      INTEGER ARRAY(100,100), SUM(256)
      INTEGER AREA, UTH,UT1,UTHF
      INTEGER AMAX(5,20)
      I=IROW
      J=JCOL
      JP=JO
      IP=IO
      MTOUCH=0
      NPT=1
      ZD=SQRT(ZMIN**2/2. )
C
C     NORMALLY, EACH SIDE OF THE 45 DEGREE TRIANGLE WOULD BE EQUAL.
C     WITH THE CAMERA BIAS, WE HAVE DIFFERENT SIZE DELTAS.
C
      JD=IFIX(ZD/1. +.5)
      ID=IFIX(ZD/1.25+.5)
C
C
C     SET DELTAS FOR THE PERPENDICULARS IN THE BOX
C
      JE=IFIX(ZMIN/1. +.5)
      IE=IFIX(ZMIN/1.25+.5)
C
C
C
      LTH=LT1-256
      UTH=LTH+NAREA
      LTHF=LTH
      UTHF=UTH
      GO TO 21
20    CONTINUE
C
C
C     IF ON EDGE OF SCAN FORGET DISTANCE CHECK
C
```

```
      IF(IM2.EQ.1)GO TO 29
      IF(JM2.EQ.1)GO TO 29
      IF(IM2.EQ.100)GO TO 29
      IF(JM2.EQ.100)GO TO 29
      AS WE TRACE EDGE OF SEGMENT, WE WANT TO PROJECT OUTWARDS FROM THE
C     CURRENT POINT. SINCE WE ARE MOVING IN THE SAME DIRECTION AROUND EDGE,
C     THIS CAN BE THOUGHT OF A PROJECTING TO THE LEFT. FOR THIS PROGRAM, THE
C     DIRECTION TO PROJECT HAS FROM THE CURRENT POINT I,J HAS BEEN DETERMINED
C     BY THE ANGLE RELATION BETWEEN IT AND THE TWO PREVIOUS POINTS I-1,J-1 AND
C     I-2,J-2. CALL THEM POINTS P, P1, AND P2. IF THE POINTS FORMED A
C     180 DEGREE ANGLE, THE A SINGLE PROJECTION AT 90 DEGREE FROM P WAS ALL
C     THAT WAS REQUIRED. SIMILARLY FOR ANY ANGLE LESS THAN 90 DEGREES.
C     FOR ANGLES GREATER THAN 90 DEGREES, YOU WOULD BE TURNING AN OUTSIDE
C     CORNER AND THUS WOULD NEED ADDITIONAL PROJECTIONS TO PROVIDE GOOD
C     COVERAGE OF POINTS AROUND THE OUTSIDE CORNER.
C     FOR INSTANCE, WITH THE OUTSIDE ANGLE OF P,P1,P2 BEING 225 DEGREES,
C     YOU WOULD HAVE 3 PROJECTIONS:
C          90 DEGREEES TO P-P1 STARTING AT P.
C          90 DEGREES TO P-P1 STARTING AT P1.
C      ?   135 DEGREES TO P-P1-P2 STARTING AT P1( SPLIT THE OUTSIDE ANGLE)
C
C     BY USING THIS TECHNIQUE, WITH EVEN MORE PROJECTIONS FOR LARGER ANGLES,
C     YOU ARE ABLE TO FORM A FAIRLY COHERENT BORDER WITH A MINIMUM OF
C     PROJECTIONS.
C
C
C
C
C
C
C     FOUND NEW POINT ALONG EDGE. FIND WHICH DIRECTION(S) TO EXTEND
C     MINIMUM VALUE AND SEE ABOUT CRITICAL DISTANCES.
C
C
C     FOR ANGLES MOVING DIAGONAL TO THE SCAN, THE DISTANCE TO PROJECT WILL
C     BE SOME COMBINATION OF THE "DELTAS" ID AND JD. FOR PROJECTIONS
C     MOVING HORIZONALLY THE DELTA IS +OR- IE. FOR PROJECTIONS
C     MOVING VERTICALLY THE DELTA IS +OR- IE.
C     WHEN A DIRECTION OF PROJECTION FROM A POINT IS DECIDED, CKPT IS
C     CALLED TO FIND IF THE PROJECTION FINDS A CRITICAL DISTANCE.
C
C     THE POINTS INVOLVED ARE I,J    IO,JO     IM2,JM2
      IF(J-JO)5,10,15
5     IF(I-IO)6,7,8
6     I1=-ID
      J1=JD
      CALL CKPT(I,J,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.GT.IO)GO TO 29
      IF(I.EQ.IM2.AND.J.NE.JM2)GO TO 29
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
```

```
C
C
C       CALL EDGE TO BREAK ARRAY INTO AREAS WITHIN LTH, UTH
C
        CALL AEDGE(LTL,LTU)
C
C
C
C
        CALL OVLOD(5,OVERS,0,IERR)
        IF(IERR.NE.1)GO TO 980
C
C
C       NOW TRY TO FIND DISTANCES BETWEEN AREAS
C
C
        CALL DISTANCE
C
        IF(NANS.EQ."N")GO TO 40
C
C
C       SET VALUES FOR AUTO MOVE TO NEXT SCAN
C
        LEFTX=LEFTX+NXSIZ
        IF(LEFTX.LT.NXSTOP)GO TO 10
C
C       MOVE UP ON SCREEN AND MAKE NEXT PASS
C
        LEFTX=NXST
        LOWERY=LOWERY+NYSIZ
        IF(LOWERY.LT.NYSTOP)GO TO 10
C
C
C       END OF SCREEN
C
 40     CONTINUE
        CALL CLOSE(5,IERR)
        CALL RESET
        STOP
980     TYPE " ERROR LOADING OVERLAY ",IERR
        STOP
        END
```

```
      GO TO 30
C
C
 20   CONTINUE
C
C     INITIALIZE DISK FILE AND UNPACK DATA
C
      CALL OVLOD(5,OVER0,0,IERR)
      IF(IERR.NE.1)GO TO 980
C
C
      CALL UNPACK
C
C
      CALL OVLOD(5,OVER1,0,IERR)
      IF(IERR.NE.1)GO TO 980
C
C
C     SMOOTH DATA IF REQUIRED
C
C
      CALL SMOOTH
C
C
 30   CONTINUE
C
C
C     CLEAR TEKTRONIX SCREEN
C
      WRITE(1,245)
 245  FORMAT("       <33><14>       ")
C
C
C
C
C
C     DETERMINE THRESHOLD
C
      CALL OVLOD(5,OVER6,0,IERR)
      IF(IERR.NE.1)GO TO 980
C
C
C
      LTU=0
C
C
      CALL CONVAL(ARRAY,LP,LTL,LTU)
C
C
C
```

```
C         FIND OUT ABOUT LIVE SCAN
C
C
12        TYPE " LIVE SCAN? Y,N"
          READ(11,100)NANS
          IF(NANS.EQ."N")GO TO 20
          IF(NANS.NE."Y")GO TO 12
C
C
C         SET UP FOR LIVE SCANS
C
C
C
100       FORMAT(S1)
C
C         INITIALIZE SCAN VALUES TO START WITH LOWER LEFT OF VISIBLE SCREEN.
C
          NXST=150
          NYST=150
          LEFTX=NXST
          LOWERY=NYST
C
C         DETERMINE FOR A GIVEN STEP SIZE HOW MANY SCANS ACROSS AND DOWN
C         THE SCREEN. SET UP VALUES AND ENDING CONDITIONS.
C
          MS=400-2*LEFTX
          STEP=1
          NUM=MS/(STEP*100)+1
          NXSIZ=MS/NUM
          NXSTOP=400-LEFTX-NXSIZ/2
          MS=500-2*LOWERY
          NUM=MS/(100*STEP)+1
          NYSIZ=MS/NUM
          NYSTOP=500-LOWERY-3*NYSIZ/2
          NUMSCN=0
10        RIGHTX=99*STEP+LEFTX
          UPPERY=99*STEP+LOWERY
          NUMSCN=NUMSCN+1
          WRITE(LP,837)NUMSCN
837       FORMAT(T30,"* * * SCAN NUMBER ",I3,"  * * *")
C
C
C
C
C         GO SCAN SCENE
C
          CALL TSPOT(LEFTX,LOWERY,ARRAY)
C
C
C
```

```
C        PROGRAM BOARD
C
C        PREPARED FOR SCI SYSTEMS, INC.
C         BY BILL POPE, TELCOM DATA CORPORATION.
C
C        THIS PROGRAM IS THE DRIVER (MAIN) ROUTINE FOR THE LIVE BOARD
C        SCANNING SYSTEM. IT PERFORMS A NUMBER OF SCANS, EACH OF 100X100 POINTS,
C        TO COVER THE AREA SEEN BY THE CAMERA. THEN SUBROUTINES ARE
C        CALLED TO DETERMINE THE FEATURES IN THE SCAN.
C        ALTERNATELY, BOARD CAN EXAMINE A SINGLE SCAN PREVIOUSLY STORED
C        ON DISC BY THE PROGRAM SCAN2.
C
C        NOTE. SINCE SUM DATA ON DISC WAS STORED BEFORE THE HARDWARE FILTER
C        WAS PLACED ON THE DIGITIZER, IN THIS MODE THERE IS AN OPTIONAL
C        DATA SMOOTHING SUBROUTINE.
C
C
C
C
         COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX,
       1 LEFTX,RIGHTX,UPPERY,LOWERY,STEP
         INTEGER RIGHTX,UPPERY,STEP
         INTEGER ARRAY(100,100), SUM(256), AMAX(5,20)
         EXTERNAL OVER0,OVER1,OVER3,OVER6
C
C
C        INITIALIZE OVERLAYS
C
         CALL OVOPN(5,"BOARD.OL",IERR)
         IF(IERR.EQ.1)GO TO 880
         TYPE " OVERLAY OPENING ERROR ",IERR
         STOP
880      CONTINUE
C
C
C
C        OPEN TEKTRONIX TERMINAL FOR PLOTTING
C
         CALL OPEN(1, "$TT01",0,IER,128)
C
         LTU=0
         LP=10
         TYPE " PRINT OR TYPE RESULTS? P,T "
         READ(11,100)NANS
         IF(NANS.NE."P")GO TO 7
         LP=12
         CALL FOPEN(12, "$LPT" )
7        CONTINUE
C
C
```

```
C
C      READ BLOCK INTO ARRAY CALLED PACK
C
       MC=1
       I=1
       IBLOCK=0
       IEND=0
       IBLK=0
       ICOUNT=0
5      READ BINARY(2) PACK
       IBLOCK=IBLOCK+1
       IBLK = PACK(1)
       IF(IBLOCK.NE.IBLK)TYPE " BLOCK COUNTS NOT EQUAL - BUG"
       IF(IBLK.EQ.20) IEND=1
C
C      UNPACK DATA
C
       DO 6 J=7,256
       ARRAY(MC,I)=IRIGHT(PACK(J),8)
       IHOLD=ILEFT(ARRAY(MC,I),8)
       ARRAY(MC,I+2)=PACK(J)-IHOLD
       I=I+4
       IF(I.LT.100)GO TO 6
       I=I-99
       IF(I.EQ.2)GO TO 6
       I=1
       MC=MC+1
6      CONTINUE
       IF(IEND.EQ.0)GO TO 5
       RETURN
990    TYPE " NOT 10,000 POINT SCAN FILE"
       STOP
       END
```

```
        DO 1 J=1,100
        DO 1 I = 1, 100
        ARRAY(I,J) = 0
1       CONTINUE
C
C       ZERO OUT SEVERAL ARRAYS FOR INITIALIZATION PURPOSES
C
        DO 2 I = 1, 256
        PACK(I) = 0
        SUM(I) = 0
        TOT(I) = 0
2       CONTINUE
C
C       ITOT SHOULD BE THE TOTAL NUMBER OF POINTS IN SUM
C       ICOUNT IS COUNTER INCREMENTED EACH TIME A DOT IS DISPLAYED
C       IBLOCK IS CURRENT BLOCK BEING PROCESSED AND IS COMPARED
C       AGAINST IBLK WHICH IS EQUAL TO PACK(1) OF THE BLOCK JUST
C       READ. THIS COMPARISON IS DONE FOR VALIDATION PURPOSES.
C       IEND IS FLAG SET AFTER BLOCK 20 IS READ TO PREVENT READING
C       PAST VALID DATA. IEND IS SET TO 1 AFTER BLOCK 20 IS READ.
        ITOT = 0
        ICOUNT = 0
        IBLOCK = 0
        IEND = 0
C
C       THIS PROGRAM SHOULD BE RUN FROM THE DGC TERMINAL.
C
        CALL OPEN(1, "$TT01",0,IER,128)
        TYPE "  ENTER FILENAME UP TO 10 CHARACTERS :"
        READ(11, 100) INAME(1)
100     FORMAT (S10)
        WRITE (LP, 101) INAME(1)
101     FORMAT ("  FILENAME IS : ", S10)
        CALL FOPEN(2, INAME, 512)
        TYPE "  FILE IS OPEN"
C
C       LOCATE TO RELATIVE BLOCK 0 FOR HISTOGRAM INFO
C
        CALL FSEEK(2, 0)
C
C       READ RELATIVE BLOCK 0 INTO SUM
C
        READ BINARY(2) SUM
C
C       CHECK TO SEE IF 10000 POINTS IN BLOCK 0
C
        DO 3 I = 1, 256
        ITOT = ITOT + SUM(I)
3       CONTINUE
        IF (ITOT. NE. 10000) GO TO 990
```

```
      OVERLAY OVER0
      SUBROUTINE UNPACK
C
C     THIS PROGRAM UNPACKS A FILE OF SCANNED DATA PRODUCED
C     BY SUBROUTINE SCAN2 AND PLACES IT IN A 100 BY 100 ARRAY.
C     CHECKS ARE MADE DURING PROCESSING TO
C     MAKE SURE THE FILE IS VALID AND THE THRESHOLDS ARE CORRECT.
C     THE NUMBER OF POINTS WHICH WILL BE PLOTTED IS GIVEN ON THE
C     CONSOLE AFTER THE THRESHOLDS ARE TYPED IN.  THIS
C     TOTAL IS THEN PROGRAMATICALLY CALCULATED AND LATER CHECKED
C     AGAINST THE FIRST NUMBER TO VALIDATE PROCESSING.
C     SUBROUTINE LOGIC IS USED TO UNPACK DATA FROM BLOCKS 2 - 21.
C     SUBROUTINE SPDOT IS USED TO PLOT ". " ON THE SCREEN OF THE
C     TEKTRONIX 4006-1 TERMINAL.
C
C
C     ARRAY WILL CONTAIN THE SCANNED VALUES OF EACH BLOCK WHICH
C     WILL BE UNPACKED BY SUBROUTINE LOGIC.
C     PACK WILL CONTAIN THE RELATIVE BLOCK 1 THRU RELATIVE
C     BLOCK 20 DATA WHICH WILL BE READ FROM THE FILENAME.
C     IHOLD WILL BE USED IN THE UNPACKING DO LOOP
C     SUM WILL HOLD THE HISTOGRAM DATA AND IS USED TO VALIDATE
C     THE FILENAME.
C     TOT WILL BE USED TO ACCUMULATE TOTALS TO CHECK AGAINST
C     THE VALUES IN SUM TO MAKE SURE EVERYTHING IS OK
C
C     SIZE IS THE MULTIPLE OF A 100 X 100 SCAN FILE
C     WHICH WILL BE DISPLAYED ON THE SCREEN.  THUS A
C     SIZE OF 4 WILL GIVE A 400 X 400 CARTOON ON THE SCREEN.
C     UTH IS THE UPPER THRESHOLD TAKEN FROM THE CONSOLE
C     STEP IS THE WAY THE IMAGE WAS ORIGINALLY SCANNED BY SCAN2
C     AND IS NEEDED IN ORDER TO UNPACK THE DATA INTO ITS RELATIVE
C     SCANNED POSITION.
C     INAME IS THE NAME OF THE FILE TO BE PROCESSED.
C
C
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
      INTEGER ARRAY(100,100), PACK(256),  SUM(256), TOT(256)
      INTEGER AMAX(5,20)
      INTEGER SIZE, UTH, UTHLOC, STEP
      DIMENSION INAME(6)
      EQUIVALENCE (AMAX,INAME)
C
C
C
C
C
C     INITIALIZE ARRAY
C
```

```
         IF(IV. GE. LTHF. AND. IV. LE UTHF)GO TO 76
         IF(NS. EQ. 8)GO TO 65
C
C
C
60       I=IP-1
         IF(I. LT. 1)GO TO 34
         J=JP+1
         IF(J. GT. 100)GO TO 25
         IV=ARRAY(I, J)
         IF(IV. GE. LTH. AND. IV. LE. UTH)GO TO 20
         IF(IV. GE. LTHF. AND. IV. LE. UTHF)GO TO 76
         IF(NS. EQ. 1) GO TO 65
         GO TO 25
C
C
C        COMPLETED CIRCLE WITH NO FIND.  SEE IF WE ARE STILL ON
C        FIRST POINT IDENTIFIED.
65       CONTINUE
         IF(IP. EQ. IROW. AND. JP. EQ. JCOL)GO TO 66
         TYPE "BOY ARE WE LOST!"
         GO TO 77
C
C
C
C
C        STILL ON FIRST POINT SO CONSIDER IT AS ISOLATED BY
C        SETTING VALUE TO ZERO
66       ARRAY(IROW, JCOL)=0
         PERIM=0. 0
         GO TO 77
C
C        END OF LOOP.  FLAG STARTING POINT AND CONTINUE
C
76       CONTINUE
         IF(I. NE. IROW. OR. J. NE. JCOL) GO TO 10
C
C
77       CONTINUE
         MTOUCH=0
         RETURN
 80      MTOUCH=1
         RETURN
         END
         IF(IP.EQ.1.AND.NS.EQ.2)GO TO 65
C
35       CONTINUE
         J=JP-1
         IF(J.LT.1)GO TO 45
         I=IP
         IV=ARRAY(I,J)
         IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
         IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
         IF(NS.EQ.4)GO TO 65
C
C
C
40       I=IP+1
         IF(I.GT.100)GO TO 55
         J=JP-1
         IF(J.LT.1)GO TO 45
         IV=ARRAY(I,J)
         IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
         IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
         IF(NS.EQ.5) GO TO 65
C
C
C
45       I=IP+1
         IF(I.GT.100)GO TO 55
         J=JP
         IV=ARRAY(I,J)
         IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
         IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
         IF(NS.EQ.6)GO TO 65
C
C
C
50       I=IP+1
         IF(I.GT.100)GO TO 55
         J=JP+1
         IF(J.GT.100)GO TO 25
         IV=ARRAY(I,J)
         IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
         IF(IV.GE.LTHF.AND.IV.LE.UTHF)GO TO 76
         IF(NS.EQ.7)GO TO 65
C
C
C
55       J=JP+1
         IF(J.GT.100)GO TO 25
         I=IP
         IV=ARRAY(I,J)
         IF(IV.GE.LTH.AND.IV.LE.UTH)GO TO 20
```

```
      GO TO 18
125   IF(I-IO) 126, 127, 128
126   NS=7
      GO TO 18
127   NS=8
      GO TO 18
128   NS=1
C
C
C     THROUGH WITH OLD POINTS.  SAVE CURRENT POINTS FOR USE NEXT TIME
C
18    CONTINUE
      IO=I
      JO=J
      GO TO (25,30,35,40,45,50,55,60),NS
C
C     NOW WE KNOW WHERE TO START, CIRCLE CLOCKWISE LOOKING FOR
C     NEXT POINT IN AREA BOUNDARY.
C
25    CONTINUE
C
C     TO PREVENT LOOPING FOR SPECIAL CASE OF ISOLATED POINT
C     ON ROW 1 (I=1), NEED FOLLOWING CHECK.
      IF(IP. EQ. 1. AND. NS. EQ. 2)GO TO 65
C
      I=IP-1
      IF(I. LT. 1)GO TO 34
      J=JP
      IV=ARRAY(I,J)
      IF(IV. GE. LTH. AND. IV. LE. UTH)GO TO 20
      IF(IV. GE. LTHF. AND. IV. LE. UTHF)GO TO 76
      IF(NS. EQ. 2)GO TO 65
C
C
C
30    CONTINUE
      I=IP-1
      IF(I. LT. 1) GO TO 35
      J=JP-1
      IF(J. LT. 1)GO  TO 45
      IV=ARRAY(I,J)
      IF(IV. GE. LTH. AND. IV. LE. UTH)GO TO 20
      IF(IV. GE. LTHF. AND. IV. LE. UTHF)GO TO 76
      IF(NS. EQ. 3)GO TO 65
C
C
C
34    CONTINUE
C
C     DITTO COMMENT AT 25
```

```
      CALL CKPT(I,J,I1,J1,MTOUCH,LTH,UTH)
      IF(IO.GT.IM2)GO TO 29
      IF(I.EQ.IM2.AND.J.NE.JM2)GO TO 29
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(J.NE.JM2)GO TO 29
      I1=0
      J1=-JE
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.LT.IO)GO TO 29
      I1=-ID
      J1=-JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(I.NE.IM2)GO TO 29
      J1=0
      I1=-IE
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      GO TO 29
29    IF(MTOUCH.NE.0)GO TO 80
C
C     HAVE WE BEEN FULL CIRCLE
C
      IF(I.EQ.IROW.AND.J.EQ.JCOL)GO TO 77
C
C
C
21    CONTINUE
      IM2=IP
      JM2=JP
      IP=I
      JP=J
C
C     FOUND AND PLOTTED A POINT.  SEARCH FOR NEXT POINT BEGINS.
C     ASSUME THE SEARCH OF ADJACENT POINTS,THE 8 POINTS TO
C     BE EXAMINED ARE NUMBERED 1 TO 8 STARTING WITH THE ONE
C     AT THE I-1,J POSITION AS 1 AND MOVING
C     CLOCKWISE.  FOR THE POINT WE ARE CURRENTLY AT, THERE
C     WILL BE A PREVIOUS POINT AS ONE OF THE 8.  OUR
C     EXAMINATION WILL START WITH THE NEXT CLOCKWISE POINT
C     AND PROCEED.
C
      IF(J-JO) 125, 120, 115
115   IF(I-IO) 118, 117, 116
116   NS=3
      GO TO 18
117   NS=4
      GO TO 18
118   NS=5
      GO TO 18
120   NS=2
      IF(I.LT.IO)NS=6
      GO TO 29
C
C
 15   IF(I-IO)16, 17, 19
C
 16   I1=ID
      J1=JD
      CALL CKPT(I, J, I1, J1, MTOUCH, LTH, UTH)
      IF(IO. LT. IM2)GO TO 29
      IF(J. EQ. JM2. AND. I. NE. IM2)GO TO 29
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      IF(I. NE. IM2)GO TO 29
      J1=0
      I1=IE
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      IF(JO. GT. JM2)GO TO 29
      I1=ID
      J1=-JD
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      IF(IO. NE. I)GO TO 29
      I1=0
      J1=-JE
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      GO TO 29
C
C
 17   J1=0
      I1=IE
      CALL CKPT(I, J, I1, J1, MTOUCH, LTH, UTH)
      IF(IO. LT. IM2)GO TO 29
      IF(I. EQ. IM2. AND. J. NE. JM2)GO TO 29
      J1=0
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      IF(IO. GT. IM2)GO TO 29
      I1=ID
      J1=-JD
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      IF(J. NE. JM2)GO TO 29
      I1=0
      J1=-JE
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      IF(I. NE. IM2)GO TO 29
      I1=-ID
      J1=-JD
      CALL CKPT(IO, JO, I1, J1, MTOUCH, LTH, UTH)
      GO TO 29
C
C
  19  I1=ID
      J1=-JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(JO.EQ.JM2)GO TO 29
      I1=0
      J1=JE
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      GO TO 29
C
C
 10   IF(I.LT.IO)GO TO 11
      I1=0
      J1=-JE
      CALL CKPT(I,J,I1,J1,MTOUCH,LTH,UTH)
      IF(JM2.LT.J)GO TO 29
      IF(J.EQ.JM2.AND.I.NE.IM2)GO TO 29
      I1=0
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.LT.IO)GO TO 29
      I1=-ID
      J1=-JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.LT.I)GO TO 29
      I1=-IE
      J1=0
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(JM2.GT.J)GO TO 29
      I1=-ID
      J1=JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      GO TO 29
C
C
  11  I1=0
      J1=JE
      CALL CKPT(I,J,I1,J1,MTOUCH,LTH,UTH)
      IF(J.LT.JM2)GO TO 29
      IF(J.EQ.JM2.AND.I.NE.IM2)GO TO 29
      I1=0
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.GT.IO)GO TO 29
      I1=ID
      J1=JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.GT.I)GO TO 29
      J1=0
      I1=IE
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(I.NE.IM2)GO TO 29
      I1=ID
      J1=-JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(JO.EQ.JM2)GO TO 29
      I1=0
      J1=JE
      CALL CKPT(IO,JO,I1.J1,MTOUCH,LTH,UTH)
      IF(IO.LT.IM2)GO TO 29
      I1=ID
      J1=JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(I.NE.IM2)GO TO 29
      J1=0
      I1=IE
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      GO TO 29
C
C
  7   J1=0
      I1=-IE
      CALL CKPT(I,J,I1,J1,MTOUCH,LTH,UTH)
      IF(IM2.LT.I)GO TO 29
      IF(I.EQ.IM2.AND.JO.LT.JM2)GO TO 29
      J1=0
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(JM2.GT.IO)GO TO 29
      I1=-ID
      J1=JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(JM2.NE.J)GO TO 29
      I1=0
      J1=JE
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(I.NE.IM2)GO TO 29
      I1=ID
      J1=JD
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      GO TO 29
C
C
   8  I1=-ID
      J1=-JD
      CALL CKPT(I,J,I1,J1,MTOUCH,LTH,UTH)
      IF(IO.GT IM2)GO TO 29
      IF(J.EQ.JM2.AND.I.NE.IM2)GO TO 29
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(I.GT.IM2)GO TO 29
      I1=-IE
      J1=0
      CALL CKPT(IO,JO,I1,J1,MTOUCH,LTH,UTH)
      IF(JO.LT.JM2)GO TO 29
      I1=-ID
      J1=JD
```

```
        .TITLE  TSPOT
;
;       PREFARED FOR SCI SYSTEMS, INC.
;        BY BILL POPE, TELCOM DATA CORPORTION.
;
;
;       THIS SUBROUTINE PROVIDES FO A RAPID 100X100 SCAN
;       AUTOMATICALLY PLACING THE VALUES INTO THE STORAGE ARRAY
;       SO ONLY ONE CALL TO THE SUBROUTINE IS NEEDED.
;
        .ENT    TSPOT
        .EXTN   .UIEX
        .EXTU
        .NREL

;       INTEGER ARRAY(100,100)
;       CALL TSPOT(ISTARTX,ISTARTY,ARRAY)

ISX     =       -167
ISY     =       ISX+1
ARRAY   =       ISY+1
IX      =       ARRAY+1
IY      =       IX+1
COUNT   =       IY+1
COLCNT  =       COUNT+1
YCNT    =       COLCNT+1
LOC     =       YCNT+1
FS.     =       LOC-ISX+1

VIDEO   =       70
```

```
      SUBROUTINE HOLE(IN,JN,IO,JO,LTH,UTH,PERIM,NBIAS,B,NBP)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C     WHEN WORKING WITH AN AREA AND FILLING IN THE CENTER ,
C     CALLED TO TRY AND TRACE THE EDGES OF THE "HOLE", FIND THE
C     PERIMETER AND AREA OF THE HOLE, AND FIND THE MINIMUM
C     SURROUNDING DISTANCE.
C
C
C     VARIABLES INCLUDE:
C
C     IN,JN           THE CURRENT POINT WITHIN THE SEGMENT WHICH SHOULD BE
C                     THE FIRST POINT ON THE EDGE OF THE HOLE.
C     IO,JO           THE POINT FOUND WHICH WAS OUTSIDE THE THRESHOLDS AND
C                     MADE US EXPECT TO FIND A HOLE.
C     LTH,UTH         THRESHOLDS OF THE CURRENT SEGMENT.
C     PERIM           PERIMETER OF THE HOLE, TO BE RETURNED.
C     NBIAS           CURRENT SEGMENT NUMBER, I.E. VALUE USED TO MARK POINTS
C     B(2,NBP)        ARRAY CONTAINING COORDINATES OF CURRENT SEGMENT.
C
C
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
      COMMON /PLT/ISTARTX,ISTARTY,MIDX,MIDY,NXSIZE,NYSIZE
      INTEGER ARRAY(100,100),SUM(256),AMAX(5,20)
      INTEGER B(2,1500),C(2,500)
      INTEGER UTH,UTHF
C
C
C     FIRST, CALL TRACE TO FIND EDGE OF HOLES.
C
      PI=3.1415926
      IM=IN
      JM=JN
      IMO=IO
      JMO=JO
C
C     SET NSPL TO MARK EDGE OF THIS HOLE. MARK IT HIGHER THAN NORMAL
C     SO THAT WE CAN DISTINGUISH IT FROM OTHER HOLES IN AREA.
C
      NSPL=768
      CALL TRACE(IN,JN,IO,JO,LTH,UTH,PERIM,NBIAS,NSPL,IMN,IMX,JMN,JMX,
     1 C,NCP)
C
      IF(PERIM.LE.20.) RETURN
C     QUIT IF SPOT LESS THAN 20 AROUND
CC
C
```

```
C
C        WE HAVE A HOLE, AND FROM SUBROUTINE TRACE KNOW THE
C        MINIMUM AND MAXIMUM VALUES FOR THE ROWS AND COLUMNS THAT
C        IT FALLS BETWEEN. NOW, FOR EACH ROW AND EACH COLUMN, FIND
C        THE MINIMUM AND MAXIMUM HOLE BOUNDARY POINT. BY TAKING
C        THE AVERAGE OF THE ROWS AND THE AVERAGE OF THE COLUMNS,
C        FIND THE CENTER OF THE HOLE(PRETENDING IT IS A CIRCLE).
C
C
         IC=0
         NC=0
C        FIND THE MARKED THRESHOLD LOWER AND UPPER LIMITS OF THE HOLE
C
         MTL=LTH+NBIAS+NSPL
         MTU=UTH+NBIAS+NSPL
         DO 6 J=JMN,JMX
C        SET THE LEFT AND RIGHT LIMITS TO ZERO
         IVS=-1
         IWS=-1
C
         IU=IMX+1
         DO 4 I=IMN,IMX
C
C        AS WE SEARCH FROM LEFT TO RIGHT USING INDEX I, WE WILL SEARCH
C        FROM RIGHT TO LEFT USIN INDEX IU.
C
         IU=IU-1
C
C        IS LEFTMOST HOLE BOUNDARY IDENTIFIED
C
         IF(IVS.NE.-1)GO TO 1
         IV=ARRAY(I,J)
         IF(IV.LT.MTL)GO TO 1
         IF(IV.GT.MTU)GO TO 1
C
C        FOUND LEFTMOST BOUNDARY (VALUE IN IV),
C        SAVE LOCATION IN IVS.
C
         IVS=I
C
C        LOOK FOR RIGHTMOST BOUNDARY
C
  1      IF(IWS.NE.-1)GO TO 2
         IW=ARRAY(IU,J)
         IF(IW.LT.MTL)GO TO 2
         IF(IW.GT.MTU)GO TO 2
         IWS=IU
 2       IF(IVS.NE.-1.AND.IWS.NE.-1)GO TO 5
  4      CONTINUE
C
```

```
C        FAILED TO FIND HOLE BOUNDARY.
C
         STOP HOLERR
C
C
C        CALCULATE SUM OF THE AVERAGE LOWER AND UPPER EDGE POINTS.
C
 5       IC=IC+(IVS+IWS)/2
         NC=NC+1
 6       CONTINUE
C
C        HAVE SUM OF EDGE POINTS AND HOW MANY FOUND.  GET AVERAGE
C
         IC=IC/NC
C
C        IC NOW HAS X COORDINATE OF THE HOLE CENTER.  WILL REPEAT
C        PROCESS ALONG COLUMNS TO GET Y COORDINATE
C
         JC=0
         NC=0
         DO 11 I=IMN,IMX
         JVS=-1
         JWS=-1
         IU=JMX+1
         DO 9 J=JMN,JMX
         IU=IU-1
         IF(JVS.NE.-1)GO TO 7
         IV=ARRAY(I,J)
         IF(IV.LT.MTL)GO TO 7
         IF(IV.GT.MTU)GO TO 7
         JVS=J
 7       IF(JWS.NE.-1)GO TO 8
         IW=ARRAY(I,IU)
         IF(IW.LT.MTL)GO TO 8
         IF(IW.GT.MTU)GO TO 8
         JWS=IU
8        CONTINUE
         IF(JVS.NE.-1.AND.JWS.NE.-1)GO TO 10
 9       CONTINUE
         STOP HOLER2
 10      JC=JC+(JVS+JWS)/2
         NC=NC+1
 11      CONTINUE
         JC=JC/NC
C
C        THE POINT IC,JC REPRESENTS THE CENTER OF OUR HOLE, IF IT WAS
C        ROUND.
C
C
C
```

```
C
C        NOW WE WANT TO FIND THE "RADIUS" FROM IC,JC TO EACH POINT ON
C        THE CIRCUMFERENCE OF THE HOLE. ALSO FIND THE MINIMUM AND MAXIMUM
C        RADIUS DEFINED. MOST IMPORTANTLY, FIND THE MINIMUM THICKNESS
C        OF THE PAD AROUND THE HOLE.
C
C        FIRST, SET THRESHOLD LIMITS FOR THE SEGMENT CONTAINING THE HOLE.
C        SECOND, SET LIMITS FOR THE BOUNDARY OF THE SEGMENT CONTAINING HOLE.
C
         MTL=LTH+NBIAS
         MTU=MTL+NAREA
         MEL=NBIAS+256
         MEU=MEL+256
C
C
         XMIN=175.
         RADTOT=0.
         RADL=0.
         RADS=400.
C
C        FOR EACH POINT ON CIRCUMFERENCE, FIND RADIUS.
C
         DO 15 NC=1,NCP
         IP=C(1,NC)
         JP=C(2,NC)
         RI=1.25*FLOAT(IP-IC)
         RJ=FLOAT(JP-JC)
         RADIUS=SQRT(RI**2+RJ**2)
C        WRITE(LP,247)IP,JP,RADIUS
C247     FORMAT(T20,"PT=",I3,",",I3,2X,"RADIUS=",F6.2)
         RADTOT=RADTOT+RADIUS
         IF(RADIUS.GT.RADL)RADL=RADIUS
         IF(RADIUS.LT.RADS)RADS=RADIUS
         IF(RADIUS.LT.1.0)GO TO 15
C
C
C        NOW WE HAVE RADIUS. TO FIND MINIMUM THICKNESS, PROJECT ALONG
C        THE RADIUS A DISTANCE OF XMIN. XMIN IS THE PREVIOUSLY FOUND
C        MINIMUM. IF STILL IN THE SEGMENT, THEN THERE IS NOT A NEW
C        MINIMUM. IF PROJECTION FALLS OUTSIDE THE MINIMUM, THEN WE
C        NEED TO POINT BY POINT MOVE OUTWARD ALONG RADIUS DIRECTION
C        UNTIL WE GET TO EDGE OF SEGMENT. THAT POINT SHOULD BE NEW
C        MINIMUM.
C
C
C        DEPENDENT ON WHETHER I OR J VECTOR LARGER, SET ABSOLUTE RATIO
C        OF ONE TO THE OTHER.
C
         IF(ABS(RJ).GT.ABS(RI))GO TO 16
         XDI=RI/ABS(RI)
```

```
      XDJ=RJ/ABS(RI)
      GO TO 17
 16   XDI=RI/ABS(RJ)
      XDJ=RJ/ABS(RJ)
 17   CONTINUE
C
C     IF THERE IS NO MINIMUM FOUND YET, USE POINT TO POINT MOVE.
C
      IF(XMIN.EQ.175.)GO TO 50
      ID=IP+IFIX(XDI*XMIN+.5)
      IF(ID.LT.1)ID=1
      IF(ID.GT.100)ID=100
      JD=JP+IFIX(XDJ*XMIN+.5)
      IF(JD.LT.1)JD=1
      IF(JD.GT.100)JD=100
C
C     ID,JD IS THE PROJECTION OF THE PREVIOUSLY FOUND MINIMUM VALUE
C     ALONG THE DIRECTION OF THE CURRENT RADIUS.
C
C     IX=ISTARTX+ID*NXSIZE
C     IY=ISTARTY-JD*NYSIZE
C     CALL SPDOT(IX,IY)
C
C     GET VALUE OF POINT TO FIND OUT IF IT IS STILL IN SEGMENT.
C
      IV=ARRAY(ID,JD)
      IF(IV.GE.LTH.AND.IV.LT.UTH)GO TO 15
      IF(IV.GE.MTL.AND.IV.LT.MTU)GO TO 15
C
C
C     PROJECTION IS OUTSIDE SEGMENT SO MUST NOW MOVE ALONG PROJECTION
C     VECTOR TIL WE FIND NEW MINIMUM.
C
C     XDI AND XDJ CONTAIN THE "DELTAS" TO MOVE IN THE I AND J
C     DIRECTION. ONE OF THEM WILL HAVE A VALUE OF 1.0 AND THE
C     OTHERS VALUE WILL BE LESS THAN 1.0. THAT WAY, WHEN WE
C     MAKE A "MOVE", WE WILL ALWAYS GO ONE DIRECTION AT LEAST
C     1.0 UNITS. BY SAVING THE "REMAINDER" NOT MOVED AND SUMMING
C     IT, WHEN THE OTHER DIRECTIONS VALUE EXCEEDS 1.0, THEN
C     WE CAN MOVE IN THAT DIRECTION ALSO. THIS WILL GIVE US
C     AS APPROXIMATE DIRECTED MOVEMENT.
C
C
 50   RMI=0.0
      RMJ=0.0
      IL=IP
      JL=JP
 51   ID=IFIX((XDI+RMI)/1
      RMI=RMI+XDI
 52   IF(ABS(RMI).LT.1.0)GO TO 53
```

```
      RMI=RMI-FLOAT(ID)
      GO TO 52
 53   JD=IFIX((XDJ+RMJ)/1
      RMJ=RMJ+XDJ
 54   IF(ABS(RMJ).LT.1.0)GO TO 55
      RMJ=RMJ-FLOAT(JD)
      GO TO 54
 55   CONTINUE
C
C     ID AND JD ARE THE NEW DELTAS.  MAKE A MOVE
C
      IL=IL+ID
      JL=JL+JD
C
C
      IF(IL.EQ.1)GO TO 15
      IF(JL.EQ.1)GO TO 15
      IF(IL.EQ.100)GO TO 15
      IF(JL.EQ.100)GO TO 15
C     PLOT SPOT JUST MOVED TO, IF DESIRED.
C     IX=ISTARTX+IL*NXSIZE
C     IY=ISTARTY-JL*NYSIZE
C     CALL SPDOT(IX,IY)
C
C     IS NEW POINT ON EDGE YET
C
      IV=ARRAY(IL,JL)
      IF(IV.GT.MEU)GO TO 51
      IF(IV.GE.MEL)GO TO 56
      IF(IV.GE.LTH.AND.IV.LT.UTH)GO TO 51
C
C     DID WE INADVERTENTLY PASS THROUGH EDGE ON DIAGONAL
C
      IF(IV.GE.NAREA)GO TO 51
C
C     YES, LOOK AT ADJACENT POINTS TO TRY AND FILD EDGE.
C
      I1=IL
      J1=JL
C
C
      IL=I1-1
      IF(IL.LT.1)GO TO 151
      IV=ARRAY(IL,JL)
      IF(IV.GE.MEL.AND.IV.LT.MEU)GO TO 56
151   JL=J1-1
      IL=I1
      IF(JL.LT.1)GO TO 152
      IV=ARRAY(IL,JL)
      IF(IV.GE.MEL.AND.IV.LT.MEU)GO TO 56
```

```
152       IL=I1+1
          JL=J1
          IF(IL.GT.100)GO TO 153
          IV=ARRAY(IL,JL)
          IF(IV.GE.MEL.AND.IV.LT.MEU)GO TO 56
153       JL=J1+1
          IL=I1
          IF(JL.GT.100)GO TO 15
          IV=ARRAY(IL,JL)
          IF(IV.LT.MEL.OR.IV.GE.MEU)GO TO 15
C
C
C
 56       CONTINUE
C         CALCULATE DISTANCE AND CHECK AGAINST MINIMUM.
C
          XDI=1.25*FLOAT(IL-IP)
          XDJ=FLOAT(JL-JP)
          XD=SQRT(XDI**2+XDJ**2)
          IF(XD.GE.XMIN)GO TO 15
          XMIN=XD
          IB=IP
          JB=JP
          IK=IL
          JK=JL
 15       CONTINUE
C
C
C
          WRITE(LP,100)NCNT,XMIN
100       FORMAT("0HOLE IN AREA ",I3/10X,"MINIMUM THICKNESS AROUND HOLE IS : ",
     1    F6.2)
C
C
C         NOW PLOT LINE ALONG MINIMUM DISTANCE
C
          CALL DOTLIN(IB,JB,IK,JK)
C
C
C
C         NOW SCAN THE AREA BETWEEN THE MIN AND MAX VALUES FOR THE HOLE.
C         FOR POINTS WITHIN THE HOLE BOUNDARY, CALCULATE AREA.  CALCULATION
C         BASED ON LOOKING AT FOUR CORNERS OF A SQURE AND COUNTING HOW MANY
C         HAVE VALUE WITHIN THE RANGE OF THE AREA SCANNED.  SINCE AREA OF HOLE
C         DEFINED BY POINTS "OUTSIDE" RANGE, EITHER ALL OR HALF A SQUARE IS
C         WITHIN HOLE AREA IF-
C            MN=4          NO PART IN HOLE
C            MN=3          HALF A SQUARE IN HOLE
C            MN<3          ALL OF SQUARE IN HOLE
C
```

```
C
C         LTHF AND UTHF SET AS LIMITS OF HOLE BOUNDARY.
          LTHF=LTH+NBIAS+NSPL
          UTHF=UTH+NBIAS+NSPL
          JM1=JMN+1
          RAREA=0.0
          IF(JM1.GT.JMX)GO TO 32
          DO 31 J=JM1,JMX
C
C         INITIALIZE FLAG TO INDICATE SCAN IS STILL OUTSIDE HOLE.
          NFL=0
          DO 30 I=IMN,IMX
          IF(NFL.GT.0)GO TO 25
C
C         LOOK FOR FIRST BOUNDARY OF HOLE
          IF(ARRAY(I,J).GE.LTHF.AND.ARRAY(I,J).LE.UTHF)GO TO 26
          GO TO 30
25        IF(ARRAY(I,J).GE.LTHF.AND.ARRAY(I,J).LE.UTHF)GO TO 26
C
C         PASSING OUT OF HOLE AREA, SET FLAG BACK
          IF(ARRAY(I,J).LT.NBIAS)GO TO 26
          NFL=0
          GO TO 27
26        NFL=1
27        NM=0
C
C         COUNT CORNERS, WATCHING OUT FOR ROW AND COLUMN 1
          IF(I.EQ.1)GO TO 30
          IF(J.EQ.1)GO TO 30
          IF(ARRAY(I,J).GE.NBIAS)NM=NM+1
          IF(ARRAY(I-1,J).GE.NBIAS)NM=NM+1
          IF(ARRAY(I-1,J-1).GE.NBIAS)NM=NM+1
          IF(ARRAY(I,J-1).GE.NBIAS)NM=NM+1
          IF(NM.EQ.3)RAREA=RAREA+0.625
          IF(NM.LT.3)RAREA=RAREA+1.25
30        CONTINUE
31        CONTINUE
C
C
C
32        CONTINUE
          WRITE(LP,107)IC,JC
107       FORMAT(10X,"HOLE CENTER-",I3,",",I3)
          RAV=RADTOT/FLOAT(NCP)
          WRITE(LP,105)RAV
105       FORMAT(10X,"AVERAGE RADIUS=",F7.2)
          WRITE(LP,106)RADS,RADL
106       FORMAT(10X,"RADIUS RANGE FROM-",F7.2," TO ",F7.2)
          WRITE(LP,101)PERIM
101       FORMAT("           CIRCUMFERENCE OF HOLE IS ",F7.2)
```

```
        WRITE(LP,102)RAREA
102     FORMAT("                   AREA OF HOLE IS ",F7.2)
        RATIO=4.*PI*RAREA/(PERIM**2)
        WRITE(LP,104)RATIO
104     FORMAT("                   4*PI*AREA/(C**2)= ",F6.2)
        WRITE(LP,103)IMN,IMX,JMN,JMX
103     FORMAT("                   LOCATION - IMN=",I3,"  IMX=",I3,"     JMN=",
     1   I3,"  JMX=",I3)
C
C
CC      NOW THE ABNORMAL NSPL VALUE AT FIRST OF SUBROUTINE MUST BE BACKED OUT.
C       THIS WILL LEAVE THE EDGE OF ALL HOLES BEING MARKED WITH A VALUE
C       OF NBIAS + 512.
C
        NSPL=-256
        DO 41 J=JMN,JMX
        DO 40 I=IMN,IMX
        IF(ARRAY(I,J).LT.LTHF)GO TO 40
        IF(ARRAY(I,J).GT.UTHF)GO TO 40
        ARRAY(I,J)=ARRAY(I,J)+NSPL
40      CONTINUE
41      CONTINUE
        RETURN
        END
```

```
      OVERLAY OVERB
      SUBROUTINE INTERNAL(B,NB,XMIN)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C     THIS SUBROUTINE DETERMINES THE MINIMUM INTERNAL THICKNESS OF A
C     PREVIOUSLY DEFINED SEGMENT. THE POINTS BELONGING TO THE SEGMENT
C     HAVE BEEN "MARKED  IN THE ARRAY WITH A VALUE UNIQUELY IDENTIFYING
C     THEM.
C
C     TO FIND THE MINIMUM, EACH POINT ON THE EDGE WILL BE EXAMINED
C     TO FIND THE NECESSARY INTERNAL DISTANCE FROM THAT POINT.
C     IF THE DOCUMENTED EXPLANATION OF THE SEARCH PATTERN IS NOT
C     SUFFICIENT, RECOMPILE THIS SUBROUTINE AFTER "UN-COMMENTING"
C     THE STATEMENTS CONCERNING PLOTTING OF SPDOT. THE SEARCH PATTERN
C      WILL THEN BE TRACED IN A VISUAL MANNER.
C
C     VARIABLES INCLUDE:
C
C     B(2,NB)          ARRAY CONTAINING THE COORDINATES OF ALL POINTS ON EDGE
C
C     XMIN             THE MINIMUM THICKNESS FOUND.
C
C
C     INITIALIZE BY SETTING SEARCH LIMITS FOR AREA
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
      INTEGER ARRAY(100,100),SUM(256),AMAX(5,20)
      COMMON/PLT/ISTARTX,ISTARTY,MIDX,MIDY,NXSIZE,NYSIZE
      INTEGER B(2,1500),UTT,UT1
C
C
C     MAKE SURE THERE IS ENOUGH PERIMETER TO REALLY CONSIDER
C     THIS AN AREA. SMALLER RUNS INTO PROBLEMS IN THE
C     METHOD USED FOR CALCULATING SLOPES.
C
      IF(NB.LE.20)GO TO 900
C
C
      PI=3.14159
      PI3=3.*PI/8.
      PI5=5.*PI/8.
C
C     IP,JP IS FIRST POINT ON EDGE. NAREA IS THE "BASE VALUE"USED TO MARK,
C     I.E. NUMBER, EDGES. CALCULATE IN NC THE VALUE THAT HAD BEEN USED TO
C     MARK THIS EDGE.
C
      IP=B(1,1)
      JP=B(2,1)
      NC=ARRAY(IP,JP)/NAREA
C
C     FROM THAT, CALCULATE THE UPPER AND LOWER LIMITS OF THIS SEGMENT.
C
      LTL=NC*NAREA
      IF(LTL.LT.1024)STOP GONG
      UTT=LTL+NAREA
      LT1=LTL+256
      UT1=LT1+255
      LH1=UT1+1
      LHU=LH1+255
      XMINR=145.
      XMIN=20000.
C
C     START WITH EACH POINT ON EDGE OF AREA FOR SEARCH PATTERN
C
      DO 75 IB=1,NB
C
C     SAVE SEARCH START POINT
C
      IP=B(1,IB)
      JP=B(2,IB)
C
C     IF POINT IS ON EDGE OF SCAN, IGNORE IT.
C
      IF(IP.EQ.1)GO TO 75
      IF(IP.EQ.100)GO TO 75
      IF(JP.EQ.1)GO TO 75
      IF(JP.EQ.100)GO TO 75
C
C     FIND SLOPE OF PERIMETER IN AREA OF THIS POINT BY LOOKING
C     AT THE LINE FROM 8 POINTS BACK TO 8 POINTS AHEAD.
C
      KP8=IB
      KM8=IB
      DO 31 M=1,8
C
C     AS WE MOVE BACKWARD ALONG POINTS IN B, IF WE REACH B(1,1) THEN
C     WILL NEED TO LOOP AROUND TO END OF B(1,NB) TO PICK UP NEXT POINT.
C     THIS IS BECAUSE B(1,1) AND B(1,NB) ARE ADJACENT POINTS WITHIN THE
C     GEOMETRY OF THE SCANNED ARRAY.
C
      KM8=KM8-1
      IF(KM8.LT.1)KM8=NB+KM8-1
C
C     IF EDGE OF SEGMENT HITS EDGE OF SCAN, THEN QUIT MOVING BACKWARD
C     BECAUSE SCAN EDGE POINTS ARE NOT RELAVENT TO SHAPE OF THE SEGMENT.
C
      IF(B(1,KM8).EQ.1)GO TO 32
      IF(B(2,KM8).EQ.1)GO TO 32
      IF(B(1,KM8).EQ.100)GO TO 32
      IF(B(2,KM8).EQ.100)GO TO 32
31    CONTINUE
32    CONTINUE
      DO 33 M=1,8
C
C     MOVE FORWARD ALONG ARRAY B WITH SAME CONSTRAINTS OBSERVED WHEN
C     MOVING BACKWARDS.
C
      KP8=KP8+1
      IF(KP8.GT.NB)KP8=KP8-NB
      IF(B(1,KP8).EQ.1)GO TO 34
      IF(B(2,KP8).EQ.1)GO TO 34
      IF(B(1,KP8).EQ.100)GO TO 34
      IF(B(2,KP8).EQ.100)GO TO 34
33    CONTINUE
34    CONTINUE
C
C     THE DIFFERENCE IN COORDINATES OF THE TWO POINTS FOUND GIVE US
C     A DIRECTED SLOPE OF THE EDGE IN THE AREA OF THE POINT I,J.
C
      IDIF=B(1,KP8)-B(1,KM8)
      JDIF=B(2,KP8)-B(2,KM8)
      XDI=FLOAT(IDIF)
      XDJ=FLOAT(JDIF)
C
C
C     IF FIRST TIME THROUGH LOOP, GO  ON TO FIND A TEST MINIMUM
C
      IF(IB.EQ.1)GO TO 49
C
C
C     BASED ON THE SLOPE JUST CALCULATED, PROJECT A POINT THE
C     MINIMUM THICKNESS ALREADY FOUND TOWARD THE INSIDE OF THE
C     AREA.  IF THAT POINT IS INSIDE THE AREA, THEN THERE IS NOT
C     A NEW MINIMUM SO WE CAN SKIP REST OF TEST FOR THAT POINT.
C
C
      IF(ABS(XDI).LT.ABS(XDJ))GO TO 43
C
C     FIND ABSOLUTE RATIO OF SMALLER DELTA TO LARGER
C
C     BY REVERSING THE I AND J VALUES, AND CHANGING THE SIGN OF THE
C     NEW I VALUE, WE ARE PROJECTING AT 90 DEGREES TO THE SLOPE.
C
      RJ=XDI/ABS(XDI)
      RI=-XDJ/ABS(XDI)
      GO TO 44
 43   RJ=XDI/ABS(XDJ)
      RI=-XDJ/ABS(XDJ)
```

```
44      I1=IP+IFIX(XMINR*RI)
        J1=JP+IFIX(XMINR*RJ)
        IF(I1.LT.1)I1=1
        IF(J1.LT.1)J1=1
        IF(I1.GT.100)I1=100
        IF(J1.GT.100)J1=100
C
C       PLOT SPOT
C
C       IXP=ISTARTX+I1*NXSIZE
C       IYP=ISTARTY-J1*NYSIZE
C       CALL SPDOT(IXP,IYP)
C
        IV=ARRAY(I1,J1)
C
C       IS POINT INSIDE AREA.  IF SO SKIP TO 75.
C
        IF(IV.GE.LTL.AND.IV.LT.UTT)GO TO 75
C
  49    CONTINUE
C
C
C
C       FIND THE ANGLE ASSOCIATED WITH THE LINE.
C
        IF(ABS(XDI).GT.1.E-4)GO TO 8
        SLOPE=3000.*FLOAT(JDIF)
        PHI=PI/2.
        IF(JDIF.LT.1)PHI=PI+PHI
        GO TO 9
 8      CONTINUE
        SLOPE=XDJ/XDI
        PHI=ATAN2(XDJ,XDI)
 9      CONTINUE
C
C       SET APPROPRIATE SEARCH ANGLE
C
        RPHI=PI3+PHI
        QPHI=PI5+PHI
C
C       FIND SLOPES OF SEARCH LINES
C
        RT=TAN(RPHI)
        QT=TAN(QPHI)
C
C       BASED ON SLOPES AND WHETHER THE DELTA I OR DELTA J SHOULD BE
C       LARGER, GENERATE CORRECT DELTAS
C
        IF(ABS(RT).GE.1.0)GO TO 10
        RDI=-1.0
```

```
      SUBROUTINE SPMOV (IXPOS,IYPOS)
C  THIS SUBROUTINE SETS THE POSITION OF THE NEXT CHARACTER TO BE
C  PRINTED ON THE TERMINAL TO IXPOS,IYPOS.
C  THE FUNCTION IS SIMILAR TO SPLIN, BUT ONLY THE FIRST (DARK)
C  VECTOR IS PLOTTED
C  THE TEKTRONIX 4006 MUST HAVE BEEN ASSIGNED TO CHANNEL 1
C  ("CALL FOPEN (1,"$TTO1")", BEFORE CALLING THE SUBROUTINE.
      IYH=288+(IYPOS/32)
C                                COMPUTE Y START HIGH BYTE
      IYL=352+MOD(IYPOS,32)
C                                COMPUTE Y START LOW BYTE
      IXH=288+(IXPOS/32)
C                                COMPUTE X START HIGH BYTE
      IXL=320+MOD(IXPOS,32)
C                                COMPUTE X START LOW BYTE
5     FORMAT (S2,Z)
C                                OUTPUT 1 CHARACTER INHIBIT CARRIAGE RETURN
6     FORMAT (S2,S2,S2,S2,Z)
C                                OUTPUT 4 CHARACTERS, INHIBIT CARRIAGE RETURN
      IGRAP=285
C                                GRAPHICS MODE CHARACTER
      WRITE (1) " "
C                                OUTPUT A SPACE FOLLOWED BY A CARRIAGE RETURN
      WRITE (1,5) IGRAP
C                                SHIFT TO GRAPHICS
      WRITE (1,6) IYH,IYL,IXH,IXL
C                                MOVE CURSOR TO START POSITION
      ISP=288
C                                NUMERICAL REPRESENTATION OF A SPACE
      IUS=287
C                                NUMERICAL REPRESENTATION OF ALPHA MODE CHARACTER
      WRITE (1,6) ISP,ISP,ISP,IUS
C                                OUTPUT SPACES TO ALLOW TIME FOR VECTOR AND THEN
      GO ALPHA MODE
      RETURN
      END
      SUBROUTINE SPLIN (IXPOS,IYPOS,IXEND,IYEND)
C   THIS SUBROUTINE CONVERTS X POSITION AND Y POSITION TO
C   THE PROPER CONSTANTS TO PRODUCE A PLOT ON A TEKTRONIX 4006.
C   THE PLOT WILL BE FROM IXPOS,IYPOS TO IXEND,IYEND.
C   THE TEKTRONIX 4006 MUST HAVE BEEN ASSIGNED TO CHANNEL 1
C   ("CALL FOPEN (1,'$TTO1')") BEFORE CALLING THE SUBROUTINE.
      IYEH=288+(IYEND/32)                  ;COMPUTE Y END HIGH BYTE
      IYH=288+(IYPOS/32)                   ;COMPUTE Y START HIGH BYTE
      IYEL=352+MOD(IYEND,32)               ;COMPUTE Y END LOW BYTE
      IYL=352+MOD(IYPOS,32)                ;COMPUTE Y START LOW BYTE
      IXEH=288+(IXEND/32)                  ;COMPUTE X END HIGH BYTE
      IXH=288+(IXPOS/32)                   ;COMPUTE X START HIGH BYTE
      IXEL=320+MOD(IXEND,32)               ;COMPUTE X END LOW BYTE
      IXL=320+MOD(IXPOS,32)                ;COMPUTE X START LOW BYTE
5     FORMAT (S2,Z)             ;OUTPUT 1 CHARACTER, INHIBIT CARRIAGE RETURN
6     FORMAT (S2,S2,S2,S2,Z)    ;OUTPUT 4 CHARACTERS, INHIBIT CARRIAGE RETURN
      IGRAP=285                 ;GRAPHICS MODE CHARACTER
      WRITE (1) " "             ;OUTPUT A SPACE FOLLOWED BY A CARRIAGE RETURN
      WRITE (1,5) IGRAP         ;SHIFT TO GRAPHICS
      WRITE (1,6) IYH,IYL,IXH,IXL       ;MOVE CURSOR TO START POSITION
      WRITE (1,6) IYEH,IYEL,IXEH,IXEL ;DRAW LINE TO END POSITION
      ISP=288           ;NUMERICAL REPRESENTATION OF A SPACE
      IUS=287           ;NUMERICAL REPRESENTATION OF ALPHA MODE CHARACTER
      WRITE (1,6) ISP,ISP,ISP,IUS       ;OUTPUT SPACES TO ALLOW TIME FOR VECTOR
      AND THEN GO ALPHA MODE
      RETURN
      END
      NTOP=ARRAY(I,J)
      GO TO 220
245   NBOT=ARRAY(I,J)
      GO TO 220
C
C
C     AT LABELS 230 AND 250, WE HAVE BEEN MOVING BETWEEN LEVELS. WHEN THE
C     VALUE OF THE SLOPE FALLS BELOW THE LIMIT NSTP, THINGS ARE "LEVELING"
C     OUT. IF THERE WAS A PREVIOUS LEVEL (NOT STARTING A SIDE OF SCAN),
C     THEN GET AVERAGE BRIGHTNESS AND SAVE.
C
230   IF(NS.LT.-NSTP)GO TO 224
      IF(NTOP.LT.0)GO TO 231
      XA=(FLOAT(NTOP+ARRAY(I,J)))/2.
      AVG=AVG+XA
      CNT=CNT+1.
231   NTOP=-1
      NBOT=-1
      GO TO 220
250   IF(NS.GT.NSTP)GO TO 224
      IF(NBOT.LT.0)GO TO 231
      XA=(FLOAT(ARRAY(I,J)+NBOT))/2.
      AVG=AVG+XA
      CNT=CNT+1.
      GO TO 231
 220  SLOPE=NS
      IF(IABS(NS).LT.NSTP)SLOPE=0
224   CONTINUE
225   CONTINUE
C
C
C     THATS ALL THE POINTS.  FIND AVERAGE FOR UPPER LIMITS.
C
      IF(CNT.LE.1.)GO TO 933
      LT=IFIX(AVG/CNT)
      WRITE(LP,100)MIN,LT
100   FORMAT("0COMBINED CONTOUR THRESHOLDS BETWEEN ", I4," AND ",I4)
      RETURN
 933  LT=0
      MIN=0
      RETURN
      END
```

```
50      IF(NS.GT.NSTP)GO TO 24
        IF(NBOT.LT.0)GO TO 31
        XA=(FLOAT(ARRAY(I,J)+NBOT))/2
        AVG=AVG+XA
        CNT=CNT+1.
        GO TO 31
 20     SLOPE=NS
        IF(IABS(NS).LT.NSTP)SLOPE=0
24      CONTINUE
25      CONTINUE
C
C
C
        DO 225 I=10,100,10
C
C       AS WE EXAMINE EACH ROW, NS REPRESENTS THE NEW SLOPE FOR THE NEXT
C       POINT TO EXAMINE WHILE SLOPE IS THE GENERAL SLOPE IN THE REGION
C       WE ARE SEARCHING. A SLOPE OF ZERO INDICATES WE ARE ON THE BOARD,
C       A NEGATIVE SLOPE INDICATES MOVEMENT TOWARDS A BRIGHTER LEVEL, AND
C       A POSITIVE SLOPE TOWARD A DARKER LEVEL.
C
C
        NS=ARRAY(I,1)-ARRAY(I,STP+1)
        SLOPE=NS
        IF(IABS(NS).LT.NSTP)SLOPE=0
        NTOP=-1
        NBOT=-1
        IB=0
        IT=0
        DO 224 J=1,LPEND
C
C       FIRST CHECK NEW POINT FOR MAXIMUM BRIGHTNESS
C
        IF(ARRAY(I,J).LT.MIN)MIN=ARRAY(I,J)
        NS=ARRAY(I,J)-ARRAY(I,J+STP)
C
C       JUMP DEPENDENT ON WHICH REGION WE HAVE BEEN IN.
C
        IF(SLOPE)230,240,250
C
C       HAVE BEEN SEARCHING ALONG LEVEL (SLOPE = 0).
C       IF THE ABSOLUTE VALUE OF THE SLOPE EXCEEDS THE LIMIT NSTP,
C       THEN CONSIDER STARTING MOVE TO NEXT LEVEL.
C
C
C       SAVE THE PRESENT VALUE IN NTOP OR NBOT DEPENDENT ON WHICH
C       DIRECTION WE ARE STARTING TO MOVE.
C
240     IF(IABS(NS).LT.NSTP)GO TO 224
        IF(NS.GT.0)GO TO 245
```

```
C
C
         NS=ARRAY(1,J)-ARRAY(STP+1,J)
         SLOPE=NS
         IF(IABS(NS).LT.NSTP)SLOPE=0
         NTOP=-1
         NBOT=-1
         IB=0
         IT=0
         DO 24 I=1,LPEND
C
C        FIRST CHECK NEW POINT FOR MAXIMUM BRIGHTNESS
C
         IF(ARRAY(I,J).GT.MAX)MAX=ARRAY(I,J)
         IF(ARRAY(I,J).LT.MIN)MIN=ARRAY(I,J)
         NS=ARRAY(I,J)-ARRAY(I+STP,J)
C
C        JUMP DEPENDENT ON WHICH REGION WE HAVE BEEN IN.
C
         IF(SLOPE)30,40,50
C
C        HAVE BEEN SEARCHING ALONG LEVEL (SLOPE = 0).
C        IF THE ABSOLUTE VALUE OF THE SLOPE EXCEEDS THE LIMIT NSTP,
C        THEN CONSIDER STARTING MOVE TO NEXT LEVEL.
C
C
C        SAVE THE PRESENT VALUE IN NTOP OR NBOT DEPENDENT ON WHICH
C        DIRECTION WE ARE STARTING TO MOVE.
C
40       IF(IABS(NS).LT.NSTP)GO TO 24
         IF(NS.GT.0)GO TO 45
         NTOP=ARRAY(I,J)
         GO TO 20
45       NBOT=ARRAY(I,J)
         GO TO 20
C
C
C        AT LABELS 30 AND 50, WE HAVE BEEN MOVING BETWEEN LEVELS. WHEN THE
C        VALUE OF THE SLOPE FALLS BELOW THE LIMIT NSTP, THINGS ARE "LEVELING"
C        OUT. IF THERE WAS A PREVIOUS LEVEL (NOT STARTING A SIDE OF SCAN),
C        THEN GET AVERAGE BRIGHTNESS AND SAVE.
C
30       IF(NS.LT.-NSTP)GO TO 24
         IF(NTOP.LT.0)GO TO 31
         XA=(FLOAT(NTOP+ARRAY(I,J)))/2.
         AVG=AVG+XA
         CNT=CNT+1.
31       NTOP=-1
         NBOT=-1
         GO TO 20
```

```
      OVERLAY OVER6
      SUBROUTINE CONVAL(ARRAY,LP,MIN,LT)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C
C     THIS SUBROUTINE IS BEING MAINTAINED IN FILE SVCONVAL AND IS
C     MOVED TO CONVAL FOR COMPILATION WHEN DESIRED.  THERE IS AN
C     ALTERNATE VERSION OF CONVAL KEPT IN FILE CONSAVE.
C
C
C     THIS SUBROUTINE TRIES A DIFFERENT METHOD FOR THRESHOLDS.
C     ASSUMMING THAT THE DATA ARRAY RECEIVED FROM A SCAN BASICALLY
C     FORMS A CONTOUR OF THE BRIGHTNESSES, CONHIST SEARCHES ONE ROW
C     AT A TIME LOOKING FOR THE CHANGES IN CONTOUR.  THE CHANGING CONTOUR
C     CAN THEN BEE SEEN AS LEVELS REPRESENTING THE BOARD, THE RUNS, AND
C     THE "BOTTOM" OF HOLES.  BY FINDING THE AVERAGE MIDPOINT BETWEEN LEVELS
C     EACH TIME THE CONTOUR MOVES, THESE MIDPOINTS CAN THEN BE
C     AVERAGED TO FIND THE "UPPER", OR DARKER, THRESHOLD.  AT THE
C     SAME TIME, THE BRIGHTEST SPOT ON THE BOARD CAN BE FOUND TO USE
C     AS THE LOWER THRESHOLD.
C
C
C
      INTEGER ARRAY(100,100),STP,SLOPE
C
C
C     THE METHOD FOR DETECTING MOVEMENT BETWEEN LEVELS IS EXAMINATION OF THE
C     SLOPE OF THE CONTOUR.  SINCE THE "X" COORDINATE WILL BE CONSTANT FOR
C     EACH COMPARISON, ONLY THE "Y" COORDINATE (DIFFERENCE IN BRIGHTNESS)
C     WILL BE EXAMINED.
C
C
C     INITIALIZE VALUES
C
      NSTP=10
      STP=4
      LPEND=100-STP
      MAX=0
      MIN=255
      CNT=0.
      AVG=0.
      DO 25 J=10,100,10
C
C     AS WE EXAMINE EACH ROW, NS REPRESENTS THE NEW SLOPE FOR THE NEXT
C     POINT TO EXAMINE WHILE SLOPE IS THE GENERAL SLOPE IN THE REGION
C     WE ARE SEARCHING.  A SLOPE OF ZERO INDICATES WE ARE ON THE BOARD,
C     A NEGATIVE SLOPE INDICATES MOVEMENT TOWARDS A BRIGHTER LEVEL, AND
C     A POSITIVE SLOPE TOWARD A DARKER LEVEL.
```

```
C         IF POINT IS IN SOME OTHER AREA GO SET FLAG.
C
          IF(IV.LT.LTH)GO TO 10
          IF(IV.GT.UTH)GO TO 10
          RETURN
C
C
 10       MTOUCH=1
C
C         FOUND CRITICAL DISTANCE. DRAW VECTOR TO SHOW AND MARK IT WITH "C".
C
          IX=ISTARTX+I*NXSIZE
          IY=ISTARTY-J*NYSIZE
          IXS=ISTARTX+IO*NXSIZE
          IYS=ISTARTY-JO*NYSIZE
          CALL SPLIN(IXS,IYS,IX,IY)
          IXE=IXS-(IXS-IX)/2
          IYE=IYS-(IYS-IY)/2
          CALL SPMOV(IXE,IYE)
          WRITE(1,100)
100       FORMAT(" C")
          RETURN
          END
```

```
      SUBROUTINE CKPT(I0,J0,I1,J1,MTOUCH,LTH,UTH)
C
C     PREPARED FOR SCI SYSTEMS, INC
C      BY BILL POPE, TELCOM DATA CORPORATION
C
C
C     SUBROUTINE CKPT PREPARES A "PRROJECTION" FROM POINT I0,J0 ALONG
C     A DELTA OF I1,J1.  IF THE NEW POINT IS WITHIN SOME AREA OTHER THAN
C     THE CURRENT ONE (CURRENT AREA HAS THRESHOLDS OF LTH,UTH),  THEN
C     THE FLAG MTOUCH IS SET INDICATING CRITICAL DISTANCE.
C
C
C     IF PROJECTION LOGIC FOUND IN TRACK IS UNCLEAR, TRY PLOTTING POINTS
C     FROM THIS SUBROUTINE TO SEE DIRECTIONS PROJECTED.
C
C
      COMMON ARRAY,SUM,LP,IMN,IMX,JMN,JMX,NCNT,NAREA,AMAX
      INTEGER ARRAY(100,100),SUM(256),AMAX(5,20)
      INTEGER UTH
C
C
      COMMON /PLT/ISTARTX,ISTARTY,MIDX,MIDY,NXSIZE,NYSIZE
C
C
C
C     FIND NEW PROJECTED POINT.
C
      I=I0+I1
      J=J0+J1
C
C
C
C     IF PROJECTION OFF SCAN, BACK UP TO EDGE.
C
      IF(I.LT.1)I=1
      IF(J.LT.1)J=1
      IF(I.GT.100)I=100
      IF(J.GT.100)J=100
C
C     IF DESIRED, PLOT PROJECTED POINT.
C
C     IX=ISTARTX+I*NXSIZE
C     IY=ISTARTY-J*NYSIZE
C     CALL SPDOT(IX,IY)
C
C
      IV=ARRAY(I,J)
C     IF POINT ON BOARD (NOT MARKED FOR ANY AREA), RETURN.
C
      IF(IV.LT.NAREA)RETURN
```

```
C
C
      RETURN
900   WRITE(LP,903)
903   FORMAT( " AREA THICKNESS INDETERMINATE")
      RETURN
      END
```

```
  85    NOPT=MPNT
        MPNT=NSTRT
        GO TO 70
C
C
C
C       OUTSIDE OF AREA, NOW WE'RE IN TROUBLE
C       SEE IF AN ADJACENT POINT IS ON BOUNDARY.  IF SO , USE IT.
C
 90     IF(IN.EQ.100)GO TO 91
        IU=ARRAY(IN+1,JN)
        IF(IU.LT.LT1.OR.IU.GT.UT1)GO TO 91
        IN=IN+1
        GO TO 95
 91     IF(JN.EQ.1)GO TO 92
        IU=ARRAY(IN,JN-1)
        IF(IU.LT.LT1.OR.IU.GT.UT1)GO TO 92
        JN=JN-1
        GO TO 95
 92     IF(JN.EQ.100)GO TO 93
        IU=ARRAY(IN,JN+1)
        IF(IU.LT.LT1.OR.IU.GT.UT1)GO TO 93
        JN=JN+1
        GO TO 95
 93     IF(IN.EQ.1)GO TO 94
        IU=ARRAY(IN-1,JN)
        IF(IU.LT.LT1.OR.IU.GT.UT1)GO TO 94
        IN=IN-1
 95     NBFLAG=1
        GO TO 64
 94     CONTINUE
C
C       IF NOT ON EDGE OF SCAN JUST KEEP TRUCKIN'
C
        IF(IN.EQ.1)GO TO 75
        IF(JN.EQ.1)GO TO 75
        IF(IN EQ.100)GO TO 75
        IF(JN EQ.100)GO TO 75
        IF(NBFLAG.EQ.0)GO TO 60
75      CONTINUE
C
C       DID WE FIND A MINIMUM
C
        IF(XMIN.EQ.20000.)GO TO 900
C
C       YES, PLOT IT
C
        X=XMIN
        XMIN=SQRT(X)
        CALL DOTLIN(IBI,IBJ,ICI,ICJ)
```

```
          ICI=IN
          ICJ=JN
   63     NBFLAG=1
  64      CONTINUE
  C
  C       IF NOT TRACKING ALONG EDGE, GO MOVE TO NEW POINT
          IF(NBFLAG.EQ.0)GO TO 60
  C
  C       HAVE WE BEEN ON EDGE BEFORE
          IF(MPNT.NE.0)GO TO 70
  C
  C       SEARCH THROUGH BOUNDARY ARRAY TO FIND WHERE WE ARE LOCATED
  C
          DO 66 M1=1,NB
          IF(IN.NE.B(1,M1))GO TO 66
          IF(JN.NE.B(2,M1))GO TO 66
          MPNT=M1
          GO TO 80
  66      CONTINUE
  C
          GO TO 75
  C
  C
  C       MOVE 1 POINT ALONG EDGE
  C
   70     MPNT=MPNT+1
          IF(MPNT.EQ.NDPT)GO TO 75
          IF(MPNT.GT.NB)GO TO 75
          IN=B(1,MPNT)
          JN=B(2,MPNT)
          GO TO 61
  C
  C
  C       SEARCH OUTWARD FROM POINT HAS FOUND AN EDGE.  FIND IF IT IS BEGINNING OR
  C       END POINT OF EDGE.
  C
    80    IF(NSTRT.NE.0)GO TO 85
          IF(MPNT.LT.IB)GO TO 75
  C
  C       SAVE START POINT OF BOUNDARY SEARCH AND SETUP TO GO FIND END POINT.
  C
          NSTRT=MPNT
          RDI=QDI
          RDJ=QDJ
          GO TO 59
  C
  C
  C       WE HAVE THE ENDPOINT ON BOUNDARY.  SAVE IT THEN GO SEARCH
  C       BOUNDARY FROM NSTRT TO NDPT.
  C
```

```
C     IF NOT, LOOP BACK FOR NEXT POINT
C
      IF(IV.LT LT1)GO TO 60
      IF(IV.GT.UT1)GO TO 60
C
C
C     IF SEARCH POINT IS WITHIN 9 POINTS OF ORIGINAL ALONG BOUNDARY OF
C     AREA, DON'T CONSIDER IT.
C
      MS=IB-9
      IF(MS.LT.1)MS=MS+NB
      DO 62 M1=1,18
      MS=MS+1
      IF(MS.GT.NB)MS=MS-NB
      IF(IN.NE.B(1,MS))GO TO 62
      IF(JN.NE.B(2,MS))GO TO 62
C
C     TOO CLOSE
      IF(NSTRT.NE.0)GO TO 64
C
C     IF JUST RUNNING ALONG EDGE, DROP POINT.
C
      MOVE=MOVE+1
      IF(MOVE.GT.5)GO TO 75
C
      GO TO 64
62    CONTINUE
      NBFLAG=1
C
C
      IF(MPNT.EQ.0)GO TO 67
C
C     MAKE SURE NOT CLOSING ON EACH OTHER.
C
      IF((MPNT-IB).LT.15)GO TO 70
      IF(IB.LT.8.AND.(NB-MPNT).LT.8)GO TO 75
C
C
C     FOUND A POINT, CHECK ITS DISTANCE
C
67    XDI=1.25*FLOAT(IN-IP)
      XDJ=FLOAT(JN-JP)
      DIST=XDI**2+XDJ**2
      IF(DIST.GT.XMIN)GO TO 64
C
C     FOUND NEW MINIMUM
      XMIN=DIST
      XMINR=SQRT(DIST)
      IBI=IP
      IBJ=JP
```

```
          RMJ=RMJ+RDJ
  57      IF(ABS(RMJ). LT. 1. 0)GO TO 58
          RMJ=RMJ-FLOAT(JD)
          GO TO 57
  58      CONTINUE
C
C
C         SET FLAG FOR NOT SEARCHING ALONG EDGE CURRENTLY
C
          NBFLAG=0
          MPNT=0
C
C         MAKE A MOVE
          IN=IN+ID
          JN=JN+JD
   61     CONTINUE
C
C         PLOT THE POINT BEING EXAMINED
C
C         IXP=ISTARTX+IN*NXSIZE
C         IYP=ISTARTY-JN*NYSIZE
C         CALL SPDOT(IXP, IYP)
C
C         IF STARTING POINT, STOP LOOP
C
          IF(IN. EQ. IP. AND. JN. EQ. JP)GO TO 75
C
C         FIND VALUE OF POINT BEING LOOKED AT
          IV=ARRAY(IN, JN)
C
C         MAKE SURE POINT IS IN AREA
C
          IF(IV. LT. LTL)GO TO 90
          IF(IV. GT. UTT)GO TO 90
C
C         IF NEW POINT ON EDGE OF SCAN, IGNORE IT
C
          IF(IN. EQ. 1)GO TO 63
          IF(JN. EQ. 1)GO TO 63
          IF(IN. EQ. 100)GO TO 63
          IF(JN. EQ. 100)GO TO 63
C
C
C         IF WE HAVE HIT A HOLE IN THIS AREA, IGNORE THIS THICKNESS.
C
          IF(IV. GE. LH1. AND. IV. LE. LHU)GO TO 75
C
C
C         FIND OUT IF IT IS ON BOUNDARY
C
```

```
      RDJ=-RT
      IF(JDIF.GE.0)GO TO 11
      RDI=1.0
      RDJ=-RDJ
      GO TO 11
10    RDI=1./RT
      RDJ=1.0
      IF(IDIF.GE.0)GO TO 11
      RDJ=-1.0
      RDI=-RDI
11    IF(ABS(QT).GE.1.0)GO TO 12
      QDI=-1.0
      QDJ=-QT
      IF(JDIF.GE.0)GO TO 13
      QDI=1.0
      QDJ=-QDJ
      GO TO 13
12    QDI=1./QT
      QDJ=1.0
      IF(IDIF.GE.0)GO TO 13
      QDJ=-1.0
      QDI=-QDI
13    CONTINUE
C
C
C
C
C     INITIALIZE
      NSTRT=0
      NDPT=0
      MPNT=0
      MOVE=0
59    CONTINUE
C
C
C
C     NOW WE ARE READY TO BEGIN SEARCH.  SET CURRENT POINTS (IN,JN) AND
C     REMAINDER FROM LAST MOVE.
C
      IN=IP
      JN=JP
      RMI=0.0
      RMJ=0.0
C
 60   ID=IFIX(RDI+RMI)/1
      RMI=RMI+RDI
 55   IF(ABS(RMI).LT.1.0)GO TO 56
      RMI=RMI-FLOAT(ID)
      GO TO 55
 56   JD=IFIX(RDJ+RMJ)/1
```

```
      SUBROUTINE DOTLIN(MX,MY,NX,NY)
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C     THIS SUBROUTINE DRAWS A DOTTED LINE BETWEEN 2 POINTS
C     GIVEN BY MX,MY AND NX,NY.  MX,MY AND NX,NY ARE GIVEN
C     AS ROW, COLUMN VALUES FROM THE 100X100 SCAN MATRIX.
C     THEY ARE CONVERTED TO SCREEN COORDINATES BASED ON THE
C     INITIALLIZATION VALUES GIVEN IN THE FOLLOWING COMMON BLOCK.
C
      COMMON /PLT/ISX,ISY,MIDX,MIDY,NXSIZE,NYSIZE
C
C
C
      IX=ISX+MX*NXSIZE
      IY=ISY-MY*NYSIZE
      JX=ISX+NX*NXSIZE
      JY=ISY-NY*NYSIZE
      IXS=IX
      IYS=IY
      MAXE=ISX+100*NXSIZE
      MINY=ISY-100*NYSIZE
C
C
C
      DASH=2.*FLOAT(NYSIZE)
C
C
C     CALCULATE DISTANCE BETWEEN POINTS
C
      XDI=1.25*FLOAT(MX-NX)
      XDJ=FLOAT(MY-NY)
      XX=XDI**2+XDJ**2
      XD=SQRT(XX)
      IF(XD.LE.1.75)GO TO 20
      SCRNXD=XD*FLOAT(NYSIZE)
C
C     IF DISTANCE BETWEEN POINTS TOO SMALL FOR DOTTED LINE,
C     REDUCE SIZE OF EACH DASH.
C
      IF(SCRNXD.LT.(3.*DASH))DASH=SCRNXD/3.
      IF(DASH.LE.0.)GO TO 20
C
C
C     CALCULATE RECTANGULAR DELTAX AND DELTAY FOR EACH DASH
C     IN THE CONNECTTING LINE.
C
      PTS=SCRNXD/DASH
      RDX=-FLOAT(IX-JX)/PTS
```

```
        RDY=-FLOAT(IY-JY)/PTS
        NDX=IFIX(RDX)
        IF(NDX.NE.0)GO TO 5
        NDX=1
        IF(RDX.LT.0.)NDX=-1
5       QDX=FLOAT(NDX)
        NDY=IFIX(RDY)
        IF(NDY.NE.0)GO TO 6
        NDY=1
        IF(RDY.LT.0.)NDY=-1
6       QDY=FLOAT(NDY)
        RMX=0.
        RMY=0.
C
C
C       TAKE OFF AND PLOT LINE
C
        IXE=IX-IFIX(RDX)
        IYE=IY-IFIX(RDY)
        NDASH=7*IFIX(DASH)/4
10      IXS=IXE+IFIX(RDX+RMX)
        IXE=IXS+IFIX(RDX)
        IYS=IYE+IFIX(RDY+RMY)
        IYE=IYS+IFIX(RDY)
        RMX=RMX+RDX
        IF(ABS(RMX).GT.ABS(QDX))RMX=RMX-QDX
        RMY=RMY+RDY
        IF(ABS(RMY).GT.ABS(QDY))RMY=RMY-QDY
        KX=IABS(IXE-JX)
        KY=IABS(IYE-JY)
        IF(KX.LT.NDASH.AND.KY.LT.NDASH)GO TO 20
        KX=IABS(IXS-JX)
        KY=IABS(IYS-JY)
        IF(KX.LT.NDASH.AND.KY.LT.NDASH)GO TO 20
        IF(IXE.LT.ISX.OR.IXE.GT.MAXE)GO TO 21
        IF(IYE.LT.MINY.OR.IYE.GT.ISY)GO TO 21
        CALL SPLIN(IXS,IYS,IXE,IYE)
        GO TO 10
C
C
C       REACHED END POINT.  CONNECT TO IT
C
20      CALL SPLIN(IXS,IYS,JX,JY)
21      CONTINUE
C
C
C       WRITE SIZE BESIDE LINE
C
        IXE=NXSIZE+IX-(IX-JX)/2
        IYE=IY-(IY-JY)/2
```

```
      CALL SPMOV(IXE,IYE)
      WRITE(1,100)XD
100   FORMAT(1X.F6.2)
      RETURN
      END
      OVERLAY OVER3
      SUBROUTINE DISTANCE
C
C     PREPARED FOR SCI SYSTEMS, INC.
C      BY BILL POPE, TELCOM DATA CORPORATION.
C
C
C     AFTER ALL SEGMENTS WITHIN THE SCANNED ARRAY ARE MARKED, DISTANCE
C     IS CALLED TO INSPECT FOR THE POSSIBLE DEFECT OF TWO SEGMENTS BEING
C     TOO CLOSE TOGETHER. FOR SOME PRESET MINIMUM CRITICAL DISTANCE (10.),
C     A VECTOR WILL BE PROJECTED OUTWARD FROM EACH POINT ON THE EDGE OF
C     THE BOUNDARY OF EACH SEGMENT IN THE SCAN. IF THE PROJECTED POINT
C     HITS ANOTHER SEGMENT, IT IS ASSUMED THAT THE SEGMENTS ARE TOO CLOSE.
C
C     ALL REQUIRED INPUT IS THROUGH COMMON.
C
C
      COMMON ARRAY,SUM,LP,IMIN,IMAX,JMIN,JMAX,NCNT,NAREA,AMAX
C
      INTEGER ARRAY(100,100),SUM(256),AMAX(5,20)
C
C
      IB=0
      IC=0
C
C     NCNT IS NUMBER OF SEGMENTS PREVIOUSLY IDENTIFIED
C
      IF(NCNT.EQ.1)GO TO 990
      WRITE(LP,200)NCNT
 200  FORMAT("0* * * THERE WERE ",I3," AREAS LOCATED.")
C
C
C     WANT TO COMPARE EACH SEGMENT TO ALL OTHER SEGMENTS SO NEED
C     NESTED LOOP.
      ZMIN=10.
      ZMIN2=ZMIN**2
C
      DO 81 NB=1,NCNT
C
      DO 80 NC=1,NCNT
C
C     IF SAME AREA, SKIP
C
      IF(NB.EQ.NC)GO TO 80
C
C     FOR THE NEW AREAS TO BE COMPARED, GET THE MINIMUM AND MAXIMUM
C     AREA COORDINATES SAVED FROM SUBROUTINE EDGE.
C
      IMINB=AMAX(1,NB)
      IMAXB=AMAX(2,NB)
      JMINB=AMAX(3,NB)
      JMAXB=AMAX(4,NB)
      IMINC=AMAX(1,NC)
      IMAXC=AMAX(2,NC)
      JMINC=AMAX(3,NC)
      JMAXC=AMAX(4,NC)
C
C
C
C     FIRST , ASSUME A BOX AROUND EACH AREA.  IF THE BOXES DON´T
C     OVERLAP, AND THE PERPENDICULARS BETWEEN THE BOXES IS GREATER
C     THAN THE MINIMUM DISTANCE FOUND SO FAR, THEN THE BOX
C     BEING INSPECTED MAY BE ELIMINATED.
C
C
C
C     SET THE DIRECTION WE CAN SEE TO 0, I. E.  BOXES OVERLAP
C
      DIRECT=0
      IF(IMAXB. GE. IMINC)GO TO 2
      XDUM=1. 25*FLOAT(IABS(IMAXB-IMINC))
      IF(XDUM. GT. ZMIN)GO TO 80
      DIRECT=3
      GO TO 8
C
  2   IF(JMAXB. GE. JMINC)GO TO 4
      XDUM=FLOAT(IABS(JMAXB-JMINC))
      IF(XDUM. GT. ZMIN)GO TO 80
      DIRECT=5
      GO TO 8
C
4     IF(JMINB. LE. JMAXC)GO TO 6
      XDUM=FLOAT(IABS(JMINB-JMAXC))
      IF(XDUM. GT. ZMIN)GO TO 80
      DIRECT=1
      GO TO 8
C
6     IF(IMINB. LE. IMAXC)GO TO 8
      XDUM=1. 25*FLOAT(IABS(IMINB-IMAXC))
      IF(XDUM. GT. ZMIN)GO TO 80
      DIRECT=7
8     CONTINUE
C
C
C     IF BOTH I COORDINATES ARE ON ONE SIDE OF AREA NB AND
C     ALSO BOTH J COORDINATES ARE ON ONE SIDE OF AREA NB, THEN
C     A QUICK EXAMINATION OF THE DISTANCE TO THE CORNERS OF
C     THE AREA MIGHT ELIMINATE IT.
C
      IF(IMAXB. GT. IMINC. OR. JMAXB. GT. JMINC)GO TO 12
      XDI=1.25*FLOAT(IMAXB-IMINC)
      XDJ=FLOAT(JMAXB-JMINC)
      XDUM=XDI**2+XDJ**2
      IF(XDUM.GE.ZMIN2)GO TO 80
      DIRECT=4
      GO TO 18
C
 12   IF(IMINB.LT.IMAXC.OR.JMAXB.GT.JMINC)GO TO 14
      XDI=1.25*FLOAT(IMINB-IMAXC)
      XDJ=FLOAT(JMAXB-JMINC)
      XDUM=XDI**2+XDJ**2
      IF(XDUM.GE.ZMIN2)GO TO 80
      DIRECT=6
      GO TO 18
C
 14   IF(IMAXB.GT.IMINC.OR.JMINB.LT.JMAXC)GO TO 16
      XDI=1.25*FLOAT(IMAXB-IMINC)
      XDJ=FLOAT(JMINB-JMAXC)
      XDUM=XDI**2+XDJ**2
      IF(XDUM.GE.ZMIN2)GO TO 80
      DIRECT=2
      GO TO 18
C
 16   IF(IMINB.LT.IMAXC.OR.JMINB.LT.JMAXC)GO TO 18
      XDI=1.25*FLOAT(IMINB-IMAXC)
      XDJ=FLOAT(JMINB-JMAXC)
      XDUM=XDI**2+XDJ**2
      IF(XDUM.GE.ZMIN2)GO TO 80
      DIRECT=8
C
 18   CONTINUE
C
C
C
C
C     BOXES TOO CLOSE FOR AREAS NB AND NC.  JUST GO CHECK AREA NB THOROUGHLY.
C
C
C     SET THE BOUNDARY LIMITS FOR NB SO WE CAN IDENTIFY AREA.
C
      LT1=NB*NAREA+256
      UT1=LT1+255
C
C
C     FIND THE MAX AND MIN COORDINATES FOR THIS AREA FROM DATA WE SAVED
C     IN AEDGE.
C
      DO 10 I=IMINB,IMAXB
      IF(ARRAY(I,JMINB).LT.LT1)GO TO 10
      IF(ARRAY(I,JMINB).LE.UT1)GO TO 11
```

```
10      CONTINUE
        TYPE " ERROR IN SUBROUTINE DISTANCE"
        STOP
11      CONTINUE
C
C
C       WE NOW HAVE A PERIMETER POINT FROM AREA NB.  INITIALIZE
C       SOME VALUES AND CALL TRACK TO CHECK FOR CRITICAL DISTANCES.
C       SUBROUTINE TRACK IS A SPINOFF FROM SUBROUTINE TRACE.  IT BASICALLY
C       TRACES THE EDGES OF A SEGMENT, BUT AS EACH POINT IS FOUND MAKES
C       THE CRITICAL DISTANCE PROJECTION.
C
C
        IO=I-1
        JO=JMINB
        CALL TRACK(I,JMINB,IO,JO,LT1,UT1,ZMIN,MTOUCH)
C
C
C       DID WE FIND A CRITICAL DISTANCE
C
        IF(MTOUCH.EQ.0)GO TO 81
C
C       YES, PRINT MESSAGE
C
        WRITE(LP,181)NB
181     FORMAT("0*********** CRITICAL DISTANCE ADJACENT TO AREA ",I3)
        GO TO 81
 80     CONTINUE
 81     CONTINUE
        RETURN
 990    WRITE(LP,991)
 991    FORMAT(" ONLY ONE AREA SO DISTANCES UNAVAILABLE.")
        RETURN
        END
```